Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Information Networks Department

DRCSN: DETECTION, REINTRODUCED, COLLABORATIVE METHOD FOR HANDLING SELFISH NODES IN MANET

A Thesis

Submitted to the Council of the College of Information Technology for the Postgraduate Studies/University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology/Information Networks

Sanaa Jafaar Hassan Ali

Supervised by

Asst. Prof. Dr. Raaid Nasur Kadham Khalil

2022 A.D **1444 A.H**

Supervisor Certification

I certify that the thesis entitled (**DRCSN: Detection, Reintroduced, Collaborative Method for Handling Selfish Nodes in MANET**) was prepared under my supervision Asst. Prof. Dr. Raaid N. Alubady at the department of Information Networks/ College of Information Technology/the University of Babylon as partial fulfillment of the requirements of the degree of master's in information technology-Information Networks.

Signature:

Supervisor Name: Asst. Prof. Dr. Raaid N. Alubady

Date: / /2022

The Head of the Department Certification

Given the available recommendations, I forward the thesis entitled "**DRCSN: Detection, Reintroduced, Collaborative Method for Handling Selfish Nodes in MANET**" for debate by the examination committee.

Signature:

Prof. Dr. Saad Talib Hasson

Head of Information Networks Department

Date: / /2022

Certification of the Examination Committee

We hereby certify that we have studied the thesis entitled (**DRCSN: Detection, Reintroduced, Collaborative Method for Handling Selfish Nodes in MANET**) presented by the student (**Sanaa Jafaar Hassan Ali**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with (**Excellen**t**)** standing as a thesis for the degree of master's in information technology-Information Networks.

Signature:
Name: Haydar Abdulameer Marhoon
Title: Asst.Prof.Dr
Date: / / 2022
(Chairman)

Signature:
Name: Ahmed M. Al-Salih
Title: Asst.Prof.Dr
Date: / / 2022
(Member)

Signature:
Name: Tariq Alwan Kadhum
Title: Lecturer
Date: / / 2022
(Member)

Signature:
Name: Raaid N. Alubady
Title: Asst.Prof.Dr
Date: / / 2022
(**Member and Supervisor**)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: Dr. Hussein Atiya Lafta
Title: Professor
Date: / / 2022
(Dean of Collage of Information Technology)

Dedication

This research paper is wholeheartedly dedicated to my husband, Kefah, for supporting me with motivation and compassion and for taking care of our child, Jafaar, who is the joy of my life. May our small family flourish.

Acknowledgements

In the name of ALLAH, Most Gracious, Most Merciful:

“Glory be to Thee! We have no knowledge but that which Thou hast taught us; surely Thou art the Knowing, the Wise”. (The Holy Qur’an - (Surah Al Baqarah 2:32))

First and foremost, I would like to praise Allah the Almighty, the Most Gracious, and the Most Merciful for providing me with the patience to endure all the hardships I faced during and before writing this thesis. May Allah’s blessing go to his Prophet Muhammad (peace be upon him), his family and his companions.

I would like to express my deepest appreciation to my supervisor (Asst. Prof. Dr. Raaid Alubady), who offered exceptional help, guidance and encouragement. Without his constructive criticism, this research would not have been possible.

I would like to express my gratitude to my father, who, although no longer with us, continues to inspire me by his example of devotion and nobleness and to my mother, the most caring person in my life. May Allah bless her with long age and good health.

I would also like to express my thanks to my loving brothers and sisters, whose words of encouragement still ring in my ears.

I am sincerely grateful to my friends, Ghufran Abdulameer and Maysam Hayder, who treated me like a sister to them and welcomed me to their houses during my study.

Abstract

Mobile Ad Hoc Networks (MANETs) are interconnected systems of wireless nodes that communicate over bandwidth-constrained wireless links. Nodes in a MANET share information with each other through many intermediate nodes. However, in some cases, nodes may not take part in the routing process properly because they are too far apart or do not have enough energy. This makes the nodes act in a selfish way. These nodes (selfish nodes) that only care about themselves, they will not send the data from other nodes. All of the current studies were presented to control selfish nodes involving finding and isolating them from the rest of the network. So far, there has been no study that tries to improve the selfish nodes and make them take part in the network's activities. This study developed a new method of dealing with selfish nodes and exploiting them to their fullest instead of isolating them. For this purpose, the proposed method is including three schemes. The first scheme, Least Energy and Least Communication Ratio is proposed in order to discover selfish nodes based on two factors: energy and the connection ratio. The second scheme is the Isolate Selfish Node scheme, which is responsible for isolating the selfish nodes according to the residual energy nodes. The third scheme that represents the main contribution of this work named Detection, Reintroduced, and Collaborative of Selfish Node, proposes an improvement of the selfish node's behavior after their detection by reducing the communication rate with selfish nodes according to their own energy. Extensive simulations have been performed using the NS-2 simulator to assess the effectiveness of the proposed method. Simulation results illustrate that the proposed method increased the throughput by (8%) and (44%) as well as packet delivery ratio by (66%) and (63%) respectively. It also decreased the packets retransmission rate by (35%) and (35%); delay by (80%) and (79); power consumption by (86%) and (82%), respectively, compared with related works (SNRRM and EBCS) in case of the number of nodes is varied. Likewise, in the case of changing the speed, the proposed method proved effective. As a final result, the proposed method achieved its purpose.

Declaration Associated with this Thesis

i. Sanaa J. H. Al-Shakarchi, Raaid Alubady. "A Survey of Selfish Nodes Detection in MANET: Solutions and Opportunities of Research", *1st Babylon International Conference on Information Technology and Science (BICITS)*, pp 1-6, 2021.

ii. Sanaa J. H. Al-Shakarchi, Raaid Alubady. "Develop a New Handling Method for Selfish Nodes in Mobile Ad-Hoc Networks ". Submitted to *Bulletin of Electrical Engineering and Informatics*. (Accepted)

Table of Content

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviation	Description
ACK	Acknowledgement
AGPS	Alkylglycerone Phosphate Synthase
AODV	Ad-hoc on Demand Distance Vector
API	Application Programming Interface
AMD	Audit based Misbehavior Detection System
CBR	Constant Bit Rate
CECAD	Cost Effective Collaborative Anomaly Detection System
CPU	Central Processing Unit
CR	Communication Ratio
DGPS	Differential Global Positioning System
DRCSN	Detection, Reintroduced and Collaborative of Selfish Node Scheme
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing Protocol
EBCS	Energy-Based Credit System
ECRCM	Erlang-based Conditional Reliability Coefficient Model
EED	End-to-End Delay
ERFBM	Exponential Reliability Factor Based Mitigation Mechanism
E-TwoAck	Enhanced Two Acknowledgment
FANET	Flying Ad-hoc Network
GAMD	Game Theoretical Method with Audit Based Misbehavior Detection
GPS	Global Positioning System
GRR	Get Route Request
ID	Identity Document
IE	Initial Energy
LELCR	Least Energy and Least Communication Ratio Scheme
IMU	Inertial Measurement Unit
ITS	Intelligent Transportation Systems
IP	Internet Protocol
IFQ	Interface Queue
Mac	Media Access Control

MANET	Mobile Ad-hoc Networks
NS-2	Network Simulator 2
NS-3	Network Simulator 3
OLSR	Optimized Link State Routing
OTcl	Object-oriented extension of Tool Command language
PARSEC	PARallel Simulation Environment for Complex Systems
PCS	Personal Communications Service
PDA	Personal Digital Assistant
PDR	Packet Delivery Ratio
PRD	Packet Rate Delivered
RERR	Route Error Message
RR	Packets Retransmission Rate
RREP	Route Reply message
RREQ	Routing request message
RTBD	Record and Trust Based Detection
SCoCoWa	Secure Collaborative Contact-based Watchdog
SHARP	Scalable Hierarchical Aggregation and Reduction
SNDA	Selfish Node Detection Algorithm
SNRRM	Selfish Node Removal Using Reputation Model
SRR	SendS Route Reply
TBUT	Token-Based Umpiring Technique
Tcl	Tool command language
TORA	Temporally Ordered Routing Algorithm
TWOACK	Two Acknowledgments
UAV	Unmanned Aerial Vehicles
VANET	Vehicular Ad-hoc Network
WRP	Wireless Routing Protocol
ZHLS	Zone-Based Hierarchical Link State Routing Protocol
ZRP	Zone Routing Protocol

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Over the past few decades, scientists and researchers have become very interested in wireless communication. With the fast improvements in wireless technology, ubiquitous computing, which maintains connectivity between mobile nodes regardless of their location, is becoming a reality more and more [1]. Mobile Ad-hoc Networks (MANETs) are wireless networks that are self-created, self-managed, and self-organized. They only work for a short time. These wireless nodes can move wherever they want and can act as a source, a destination, or an intermediate router in a network. This means that they can send and receive information. The network communication working is affected by the movement of nodes [2].

These networks seem to be easy to use, regardless of where they are located geographically. Such networks may exist at any time and from any location, and they can deliver services in areas where the potential for infrastructure networks seems to be remote. This is explained by the fact that short-range communication may not need the utilization of infrastructure. In contrast to cellular networks, there is no central controlling unit in a MANET, which distinguishes it from them. This unique characteristic has attracted its use in the fields of defense, emergency response, healthcare, combined or collaborative networks, and other fields [1].

Packet routing is an important part of MANETs. For each pair of nodes that are not next to each other, the intermediate nodes must send data packets to the destination nodes. Due to how dynamic and spread out the nodes are, energy use is one of the biggest problems in MANET. Energy use is especially important since all node area units run on batteries [3]. As a result, opportunistic routing algorithms make the assumption that each node will forward each packet it receives. This has not always been the case, though, because some nodes use the resources of other nodes to communicate and will

not forward packets from other nodes within their radio spectrum. These nodes are said to be selfish or act badly in some way [4]. Even if only one node fails, the whole network can be affected. If a node does not have a lot of energy, it could operate in a selfish manner or create issues by dropping packets [5].

They do not need to use their energy, CPU, or bandwidth to send the data. In point of fact, each and every node that makes up a MANET has the potential to exhibit a selfish personality. In order to maximize revenues from network resources, but hesitant to share its resources with other nodes. In a situation where every node is required to send packets to its neighbours, a few selfish nodes deny doing so. Except for packets intended for them, these nodes block all traffic. For their own purposes, these nodes consume the network and its resources without providing service back [6]. Selfish nodes hurt the performance of the network in ways like network partitioning, fewer data availability, shorter network life, reduced throughput, and more packets being dropped [7].

1.2 Related Works

Josh Kumar et al. [8] demonstrated that selfish MANET nodes might be identified and eliminated using the Token-Based Umpiring Technique (TBUT). Each new node in the TBUT network receives a unique token upon activation. NodeID, status, and reputation are the three fields that comprise the token's three-field structure. Token status bit "1" indicates a "red flag" in the protocol, which prevents the node from participating in any network activities. To put it another way, it's not allowed to engage in any network activity if a node's reputation value is "1," which indicates "negative repute." The simulation model based on QualNet 5.0 is built up in the practical portion. 100 wireless mobile nodes that create an ad-hoc wireless network across a rectangular (1000m x 1000m) flat surface was used to evaluate

performance. TBUT considerably increases performance in all parameters, including packet delivery ratio and control overhead. Real-world applications may benefit from TBUT's enhanced security and network performance, according to security studies and testing findings.

Janakiraman and Rajendiran [9] proposed the Erlang-based Conditional Reliability Coefficient Model (ECRCM). The Erlang-based Conditional Reliability Co-efficient (ECRC) is used to quantify the influence of selfish nodes on the network's resilience, and this model uses this factor to define the extent of the damage. Using a conditional probabilistic technique, can not only determine the dependability of individual nodes but also the whole network's resiliency. Network simulator ns-2.26 was used to do an exhaustive simulation of the suggested model. At any one time, up to 100 mobile nodes may be set up throughout an area of 1,000 x 1,000 square metres (m2). According to simulation studies, ECRC Model has a successful detection rate of 28%, which is regarded as a significant achievement. In addition, may use this model to arrive at a saddle point for selfish detection of 0.3 and a resilience threshold of 0.4 using this model.

Sengathir and Manoharan [10] argued that, according to Exponential Reliability Factor Based Mitigation (ERFBM), it is possible to detect the selfish behaviour of nodes using the available energy metrics and then separate them from the routing route by reinforcing their selfishness in order to ensure accurate data dissemination. There are two ways to method the isolation of selfish nodes. When it comes to mobile nodes, there is a considerable likelihood that a cooperative node will become a selfish node if there is a lack of accessible energy. Second, the exponential dependability factor may be used to determine whether or not the nodes' selfishness should be reconfirmed, and the choice to isolate them from the routing route is then taken into account. Ns-2.26 is used to conduct extensive ERFBM simulation

tests in the practical phase. There are 100 mobile nodes in the simulated network, which is spread out across 1000 by 1000 pixel area. Findings show that the proposed ERFBM isolates selfish nodes more quickly and improves network performance by lowering both control overhead and overall overhead, according to simulation results. The planned ERFBM strategy also helps to cut down on energy use.

Prasath and Scholar [11] proposed a new approach called Record and Trust Based Detection (RTBD) Technique with Collaborative Watchdog. A packet-dropping detection mechanism and a strategy to mitigate selfish nodes are part of the proposed solution. Each time a neighbour node communicates with the selfish node, the neighbouring node receives a trust report detailing the node's prior communications. An adjacent node may tell whether the selfish node is dropping packets based on this report. Using the trust report, the neighbouring node is able to determine which node has misreported packets. In order to avoid detection, a selfish node may submit a bogus record. To improve MANET performance, the recommended RTBD approach may be used. The packet delivery ratio and detection ratio both improve dramatically as a result of this. Moreover, it decreases the overhead, latency, and packet dropping ratio.

Bama and Indir [12] indicated that Chord Algorithm, when a node in a networking group is assigned a key, the key is sent to a neighbour node in the networking group. Sending a packet from one node to another using that chord key is the next step. Whenever a MANET chord node is given a unique ID key, it will broadcast that unique ID key to all other nodes in that network group. Any intermediary nodes in the network that refuse to pass the key to another node will be identified as selfish nodes by the Key sender node in the network, and that node will broadcast that information to neighbour nodes in

the buddy list group and update its node frame. The selfish node in such a network is easy to see since use the watchdog activity monitoring approach.

Sayyar et al., [13] used the AODV routing protocol to construct an Enhanced TWOACK protocol. In the E-TwoAck scheme, nodes resend the data packet twice and wait for an acknowledgement before giving up on the current route. This is done to spot attacks by selfish nodes. This protocol also uses a discard ratio calculation on the current route to identify selective forwarding attacks. ETwoAck and TWOACK systems were implemented via AODV and DSR protocols, respectively, using NS-2 (version 2.35). A total of 100 nodes were employed in this simulation. The AODV+ETwoAck method gives up to a 90% packet delivery ratio in the presence of 40% misbehaving nodes in the network, compared to a 50% packet delivery ratio for DSR+TWOACK. In the presence of 40% misbehaving nodes, the suggested scheme's routing overhead and end-to-end latency are superior to DSR+TWOACK.

Patil [14] pointed out that in order to identify the selfish node, a Secure Collaborative Contact-based Watchdog (SCoCoWa) should use a technique that compares a node's Id and mac address to build a hash value that is checked by the sender. If the hash value does not fall within a certain range, SCoCoWa determines that this is a misbehaving node. Detection of selfishness in the routing route is recorded in the background by a watchdog mechanism in every node. When a node makes contact with a sender node, the sender node transmits the previously stored information about the other node's activity. The number of nodes in the simulator is 40, and the number of selfish nodes is 5 in this technique. Using this approach ensures that information is sent in a proper manner to the recipient without tampering with its substance. In mobile Adhoc networks, where packet forwarding relies

heavily on collaboration, this ensures the security and confidentiality of data sent through the network.

Narayanan et al. [15] used Game theoretical approach with an Audit-based Misbehavior Detection System (GAMD) to identify selfish nodes and isolate them from the network. In addition to finding selfish nodes, the proposed technique also detects helpful and evil nodes. Integration of the game theoretical method with AMD resulted in lower costs and quicker detection of both selective and continuous packet drops. There were 50 simulated mobile nodes in this experiment. The findings of the simulation reveal that there is no delay, control overhead, packet loss, or an increase in the percentage of packets delivered.

Mubeen and Johar [16] completed the identification of a selfish node using an Energy-Based Credit System (EBCS). The selfish node will be eliminated from packet transmission in this scheme. The energy-based credit system identifies the typical nodes and credits them with energy. Checking the threshold energy level of all nodes accused of selfishness is done through the energy-based credit system NS-2, and ten nodes are used to model this system. Ad-hoc networks operate better when powered by a payment system based on energy.

Musthafa et al. [17] proposed an Selfish Node Detection Algorithm (SNDA). Where a node's desire to participate in routing operations is tested using the SNDA, based on how many routing packets the node in the network has lost over time, a threshold value ranging from 0 to *t* will be determined. A node is deemed gentle or somewhat selfish if the selfish threshold (ST) is lower than the threshold value (*t*). If the selfish threshold (ST) is greater than or equal to t, the node is deemed selfish. Nodes' reluctance to forward packets in the network is represented by the *t* value. Network Simulator (NS-2) version 2.33 is used for simulation in the practical section. A flat area of 1500

x 300 metres is used to simulate 50 to 100 nodes. With a 512-byte packet size and a data rate of four packets per second, UDP traffic with a Constant Bit Rate (CBR) is employed. This approach is able to quickly and effectively identify and isolate the selfish node from the network.

Mangayarkarasi and Manikandan [18] developed a Cost-Effective Collaborative Anomaly Detection System (CECAD) for MANET selfish node assaults. An anomaly detection module is first executed on a set of monitoring nodes that the system has selected. When a source node has to share information with a destination node, it depends on the monitoring nodes to do so cooperatively. The data collection component is in charge of gathering information about each node from packets of control and data. The fuzzy logic decision is used to identify the nodes that are either weakly or significantly suspicious. False positives and missed detections are less likely since the attacks are validated by a cooperative exchange of monitoring nodes. NS-2 and 60 to 140 nodes are used to replicate this system, with 10% of the nodes being attackers. According to the findings of the simulations, the CECAD system has reduced detection latency while also increasing detection precision.

Ponnusamya, Senthilkumarb and Manikandan [19] proposed the Selfish Node Removal Using Reputation Model (SNRRM) strategy. It is necessary to assess a node's reputation in order to exclude selfish nodes from the routing process. The current energy level of a node and the communication ratio of that node are both taken into consideration when determining that node's reputation. The sender node is where the communication is initiated when the source node (shown by an 'S') and the destination node (indicated by a 'D') have both been specified. If both 'S' and 'D' fall inside the communication range, the node will merely verify the reputation value of '***S***', and if it is a match, the transmission procedure will be completed, and the system will be

updated. If both 'S' and '***D***' do not fall within the communication range, then 'S' will send control packets to its neighbours and wait for reply messages to arrive in its inbox. In this case, the checks on reputation are a little bit difficult due to the fact that selfish nodes do not readily reply to the messages that are delivered. As a result, the communication ratio between the nodes is calculated using the request message that was sent out and the reply message that was brought in. 2.35 is the value that is utilised in this simulation, and there are a hundred nodes involved. The simulation result for the suggested model is given here, and it demonstrates that the efficiency that is reached is higher in terms of reputation ratio and delivery rates.

As a summary, Table 1.1 highlights the main points for each related study.

Table 1.1 Summary of Related Works

Authors and Ref.	Methods	Based on	Strategy	Routing Protocol	Software simulation	Max No. of Nodes	Performance Evaluation
Josh Kumar et al., [8]	TBUT	Reputation	Prevents the selfish node from participating in any network activities	AODV	QualNet 5.0	100	Packet Delivery Ratio, Failure to Detect (false negative) Probability and Control Overhead.
Janakiraman and Rajendiran [9]	ECRCM	Reputation	Selfish node in the routing path is isolated	AODV	NS2.26	100	Packet Delivery Ratio, Throughput, Control Overhead and Total Overhead
Sengathir and Manoharan [10]	ERFBM	Reputation	Detecting and isolating selfish nodes	AODV	NS2.26	100	Packet Delivery Ratio, Throughput, Control Overhead, Total Overhead, Detection Rate, False Positive Rate
Prasath and Scholar [11]	RTBD Technique with Collaborative Watchdog	Reputation	Detecting and isolating selfish nodes	N/A	N/A	N/A	Packet delivery Ratio and Detection ratio

Bama and Indir [12]	Chord Algorithm	Reputation	Detect and avoid selfish node	N/A	N/A	N/A	Overhead, Scalability, Load Balance and Time of Selfish Nodes Detection
Sayyar et al., [13]	E-TwoAck	Acknowledge	Detect selfish node	AODV	NS2.35	100	Packet Delivery Ratio, Routing Overhead and End-to-End Delay
Patil [14]	SCoCoWa	Reputation	Detect and choose the optimal path	AODV	Glomosim	40	False Positive. False Negative, Throughput, Packet Delivery Ratio, Delay and Routing Overhead
Narayanan et al. [15]	GAMD	Reputation	identify selfish nodes and isolate them from the network	DSR	NS2	50	Average Delay, Control Overhead, Packet Delivery Ratio and Packet loss
Mubeen and Johar [16]	EBCS	Credit	Detect and eliminate selfish node	DSR	NS2	10	PDR Throughput[kbps] End-to-End Delay[ms]
Musthafa et al., [17]	SNDA	Reputation	identifying and isolation of selfish node from the network	AODV	NS2.33	100	Packet Delivery Ratio (PDR), Throughput and End-to-End Delay (EED)
Mangayarkarasi et al [18]	CECAD	Reputation	Detect and choose the optimal path	N/A	NS2	140	Packet Delivery Ratio, Remaining Battery Energy, Detection Delay and Detection Accuracy
Ponnusamy et al [19]	SNRRM	Reputation	Detect selfish node and choose reliable rout	N/A	NS2.35	100	Packet Rate Delivered (PRD), Reputation Ratio and Energy Consumption

1.3 Problem Statement

Traditional MANET protocols presume that all mobile nodes must cooperate together in order for the network to work. Selfish node behaviour might emerge if nodes refuse to cooperate since it is a costly activity. Overall network performance might be negatively affected by this. In the existing methods, the node may be self-centred. There may be a considerable influence on the network's performance when nodes behave egoistically since they are reluctant to transfer packets from their source to their destination [20]. The healthiness (unselfish) nodes' throughput, latency, and packet delivery rates are all significantly lowered as a result of their behaviour [2]. Selfish node identification is not a simple task, although various methodes such as [12][13][21] have been successful in preventing them from accessing any network resources. A MANET's performance may not be improved merely by identifying and isolating selfish nodes. There is currently no way to convince them to cooperate until they have expended their energy. Therefore, the proposed method is to resolve a solution to the problem of selfish nodes based on making them collaborative in the network. Hence, greatly improving the network performance.

1.4 Research Questions

The research tries to answer the following questions about addressing the impact of selfish nodes on MANET's performance:

i. What are the parameters and their relation to MANET when it is presented with selfish nodes?

ii. How could early detect the selfish nodes in MANET?

iii. How should the proposed scheme adapt the data rate and motivate the selfish nodes to have cooperated according to network status?

iv. How can develop a new extension of the AODV protocol to moderate the selfishness discovery process?

v. How can validate the proposed method to ensure its usability and performance?

1.5 Research Objectives

The aim of this research is to develop a selfishness node handling method. The proposed method has the ability to detect selfish nodes and motivate them to cooperate in data delivery in order to increase the performance of the network. This aim can be reached even more with the following research objectives:

i. To investigate how selfish nodes can affect performance in a MANET scenario with different parameters. Including the number and speed of the nodes, which may affect on energy and communication ratio.

ii. To design a scheme for identifying selfish nodes early based on their energy and communication rate to reduce the detection time of selfish nodes.

iii. To build a model that can deal with the behaviour of selfish nodes by adjusting the threshold according to network status and motivating them to cooperate as much as possible.

iv. To develop a selfishness node handling method for MANET based on standard AODV protocol and modify the required parameters.

v. To evaluate the performance of the proposed method in comparison with available solutions in a simulated network environment.

1.6 Research Scope and Significance

The scope of this study includes implementing a method that has the ability to detect a selfish node based on the energy of nodes and their communication ratio. In addition, handling them by reducing the number of requests sent to the selfish nodes. While the main importance of this research lies in improving the performance of the MANET network in terms of packet delivery ratio, packets retransmission rate, throughput, end-to-end delay, and power consumption.

1.7 Organization of the Thesis

There are five chapters in the thesis. Each chapter starts with a short introduction that says what the chapter is about and what its main points are. The following is a summary of each chapter:

Chapter One gives a general overview of the research area. It shows the problems with this study and emphasizes how important the study is.

Chapter Two explains the theoretical side. It gives an overall look at Wireless Ad-hoc Networks, selfish nodes, and the AODV protocol. It gives lists of examples and diagrams to help the reader fully understand the subject of the thesis and how it works.

Chapter Three gives a general overview of the details of the methodology and research design that are used to reach the research goals. In addition, this chapter presents a number of different methodes that, when combined, may be used to develop and build the suggested method.

In **Chapter Four**, simulations are used to show how the whole proposed method is evaluated. In this chapter, the method is looked at in different situations. This chapter also shows how the proposed method compares to the current solution from a theoretical and graphical point of view.

Chapter Five is the last chapter. It talks about how the main research goals are met and what the main contributions of the thesis are. Based on what this study found, it also talks about some possible directions for future studies.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Introduction

Wireless mobile networks is growing increasingly popular with the public because they enable individuals to access information and connect with one another at any time and from any place. Traditional mobile wireless communications need a wired infrastructure (like Internet). Mobile Ad-hoc Networks, often known as MANETs, have become an important component of next-generation wireless networking technologies during the course of the last decade. Selfishness is one of the many forms of inappropriate behaviour that a node in a MANET network might display. A selfish node is one that seeks to protect its own resources while simultaneously using the services of other nodes and devouring the resources other nodes possess. There is a possibility that the overall performance of the network may suffer.

This chapter represents the theoretical part of this work. In light of what has been discussed up to this point, the following is the structure of this chapter: the background of ad-hoc wireless networks and the many classifications of these networks is discussed in Section 2.2. Section 2.3 goes into detail MANET architecture, including its classification, advantages, and limitations, as well as its routing protocols. Ad-hoc on-demand distance vector is presented in Section 2.4. The selfish node in MANET that is related to this research scope is described in Section 2.5, including it is behaviours, issues, and techniques to detect it in MANET. The simulation environment is illustrated in Section 2.6. Finally, the chapter ends with Section 2.7, which goes into details about performance metrics that have been used in this work.

2.2 Wireless Ad-hoc Networks

A wireless ad-hoc network's backbone is made up of mobile nodes equipped with wireless transceivers. It requires no pre-existing infrastructure to be constructed inside it. The mobile nodes of the network use wireless transceivers to exchange data; if the data is beyond the range of the

transceivers, it may be relayed by other intermediate nodes. Military and sensor networks, disaster relief, and emergency response may all benefit from wireless networks in situations when a wired network is not viable [22].

2.2.1 Classification of Wireless Ad-hoc Networks

As seen in Figure 2.1, ad-hoc wireless networks are classified as MANET, Vehicular Ad-hoc Network (VANET), and Flying Ad-hoc Network (FANET) based on their usage, deployment, communication, and mission objectives [23].

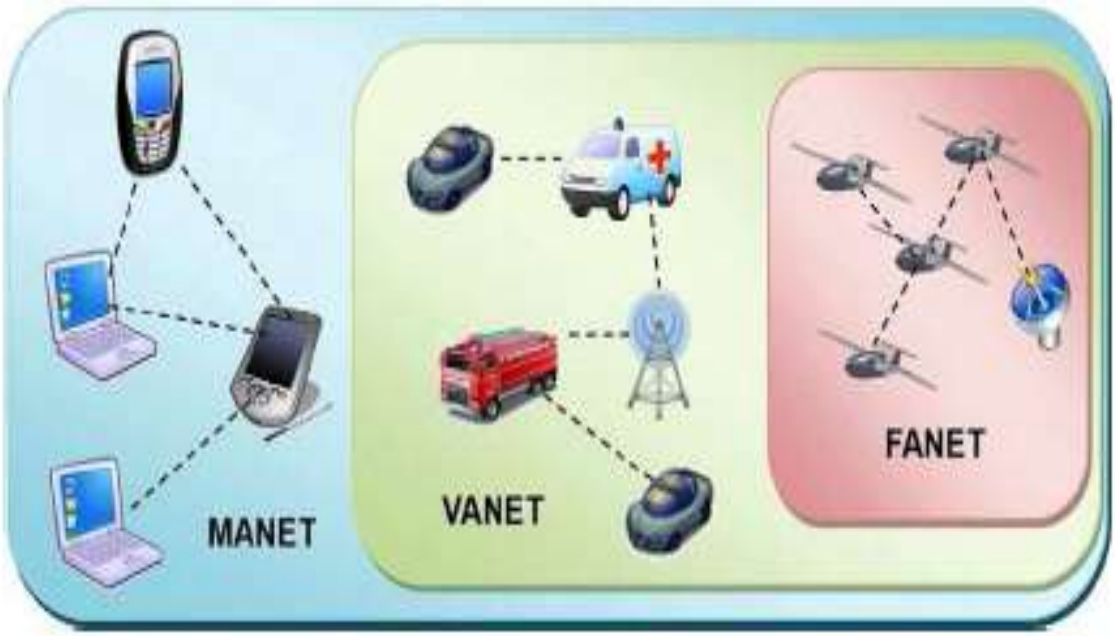

Figure 2.1 MANET,VANET and FANET Architectures [23]

2.2.1.1 Mobile Ad-hoc Network

Mobile devices link to a base station and the wired infrastructure through a single-hop wireless radio connection. When it comes to MANET, there is no existing infrastructure required. Single and multiple hop paths are used by nodes in MANET to interact with one another. As the central node links other nodes in a network, routers serve as an intermediary node. MANET nodes are both hosts and routers, as a result. Changing the number and location of nodes

affects how routing paths are constructed. The network's topology might change quickly and unexpectedly [24].

In MANET, a network of mobile hosts, does not have a defined architecture for central administration (e.g., base stations or access points). Wireless communication among mobile hosts' antennas occurs. As a result of variables such as radio power limitation and channel utilization, the mobile host may not be able to communicate directly with other hosts. It is necessary for packets to be routed via many intermediate sites before they reach their final destination in a multi-hop scenario. This necessitates that each MANET node be a router [25].

According to [24], MANET communication differs from wired networks in the following ways:

- This kind of network has bandwidth and battery power constraints. Algorithms and protocols must be developed in such a manner as to save bandwidth and energy. As a result of their restricted processing power, wireless devices generally use low-capacity computer components (processors, memory, I/O devices). Designing communication protocols that use as little processing and storage space as possible is thus critical.
- The communication architecture is constantly evolving due to the mobility of the nodes. When primary routing paths are broken, new ones are formed dynamically.
- Unlike a wired network, a wireless network allows all nodes in the transmission range of one node to hear the packets simultaneously.

2.2.1.2 Vehicular Ad-hoc Network

VANET, allow automobiles and roadside devices to communicate with one another wirelessly. Incorporating new-generation wireless networking

capabilities into automobiles is a developing technology. VANET's primary goal is to give mobile users on-the-go access through other networks at home or at work, as well as effective vehicle-to-vehicle communications to support Intelligent Transportation Systems (ITS). Co-operative traffic monitoring, traffic flow management (including blind crossings), accident avoidance, and real-time detour route calculation are just a few of the many ITS uses[26].

VANETs are an instance of MANETs. where mobility restrictions and driver behaviour all contribute to the distinctive features of VANET, these traits have significant effects on how these networks are designed, while MANETs function fundamentally differently. MANETs lack a permanent infrastructure and rely on regular nodes to handle network administration and message routing tasks. The following are the main differences [27]:

- The topology of a VANET is constantly changing, making it difficult to keep up.
- The VANET is prone to frequent fragmentation at high deployment rates.
- The VANET's effective network diameter is small.
- Battery life is a major issue in sensors and other types of mobile networks, yet, there are no significant power constraints here.
- Potentially large-scale.
- The network's density may shift.
- As the signals are received and processed by the drivers, the topology of the network may be altered. To put it another way, the message content may change the structure of a network.

2.2.1.3 Flying Ad-hoc Network

Aerial vehicles that can fly independently or be operated remotely from afar without the need for a pilot are now possible because of advances in electronic, sensor, and communication technology. Unmanned Aerial

Vehicles (UAVs) have a lot of potential for both military and civilian uses because they are adaptable, flexible, easy to install and have low operating costs. Some examples of military and civilian uses are search and destroy operations, border surveillance, wildfire management, relays for ad-hoc networks, wind estimation, disaster monitoring, remote sensing, and traffic monitoring. Even though single-UAV systems have been used for decades, a swarm of small UAVs may have a number of benefits. Multi-UAV system design, on the other hand, has its own challenges, on the other hand presents unique challenges, one of which is the communication between the various aircraft [28]. Table 2.1 shows comparison between MANET, VANET and FANET.

Table 2.1 Comparison Between MANET, VANET and FANET [23]

Features	MANET	VANET	FANET
Node mobility	Low	High	Very high
Mobility model	Random	Regular	Regular for routes that have already been specified, but specialized mobility models for autonomous multi-UAV systems
Node density	Low	High	Very high
Topology change	Slow	Fast	Fast
Radio propagation model	Close to ground, Los is not always available.	Close to ground, Los is not always available.	Most of the time, Los is available high above the ground.
Power consumption	Energy-efficient protocols	Not needed	Small UAVs do not need to be energy efficient, but mini UAVs do.

Network life time	Limited	High	High
Computational power	GPS	GPS,DGPS,AGPS	DGPS,IMU,GPS,AGPS
Localization	Low	High	Very high

2.3 Mobile Ad-hoc Network Architecture

2.3.1 Characteristics of MANET

Laptops, smartphones, tablets, and other mobile computers are all examples of the term "personal digital assistants." In any transitory network design, nodes may self-organize. Infrastructure does not exist; hence there is no clear boundary. Here are some of the most important MANET characteristics [29]:

- No centralized server, specialized equipment, or fixed routers are required: MANET does not need any kind of infrastructure. Inter-node communication is only possible through wireless.
- Wireless connection makes the Mobile Ad-hoc Network exposed to a wide range of attacks. Due to wireless nodes' limited power supply and mobility, communication participants in a mobile ad-hoc network encounter obstacles.
- Nodes may travel from one point to another on their own, making them "mobile nodes," as the name implies. To find a certain node in the network, have to keep up with the continually shifting topology. It is quite simple to enter or leave the radio range of another node. This means that the routing information nodes carry is always evolving as their motions become more unpredictable.
- With so much power, there is a limit to what can be done. The mobile hosts are lightweight and portable. A battery or other limited power

source is all they have. In the event of a network split, it might target specific node batteries and disconnect them. Mobile nodes may also be targeted by attackers' that seek to activate them aggressively, causing them to drain their batteries before they have a chance to do so

- The network architecture may change quickly and unpredictably at any time, and it can contain both bidirectional and unidirectional links.
- It features a decentralized architecture, with all mobile nodes acting as routers and all wireless devices connecting to one another. Self-configuration networks, such as MANETs, are networks in which the nodes themselves perform network functions, such as discovering the network topology and sending messages.
- Wireless connections have a much lower capacity than corded ones. Due to multiple access, fading, noise, and interference, wireless connection capacity may decrease with time. The radio's maximum transmission capacity may be smaller than the actual throughput.
- A MANET's nodes might be in this scenario if it depends on batteries or other finite energy sources to operate. According to method design, the most important aspect is likely energy conservation.
- The smaller scale of a mobile wireless network makes it more susceptible to physical security threats than a fixed-wire network. The dangers of eavesdropping, spoofing, and denial of service attacks are becoming increasingly prevalent. Protocol design for routing is guided by basic assumptions and performance challenges that go beyond the fixed internet's higher-speed, semi-static architecture.

2.3.2 Classification of MANET

There is no standard way to classify ad-hoc networks in the scientific literature. To classify networks, a number of things are taken into

consideration: the configuration of the nodes, the topology of the network, and the communication protocol (single or multiple hops).

2.3.2.1 Classification based on the Communication

This kind of network allows either a one-way or a multi-way communication, depending on how it is built up [30].

A. **Single-Hop Ad-hoc Network:** It is possible for nodes to interact directly with each other. As the name suggests, single-hop, or point-to-point ad-hoc networks, allow nodes to interact directly with each other since they are all located within a small, mutually exclusive range. It is possible to relocate the network as a whole, but the individual nodes must stay within the range of all other nodes. This would not change the communication connections.

B. **Multihop Ad-hoc Network:** This class of ad-hoc networks is the most thoroughly studied in the literature. In contrast to the first class, some nodes are too far apart to interact with each other directly. Because of this, the traffic between these communication endpoints must be routed through other nodes. The nodes are assumed to be movable in this class as well. Node mobility is a major challenge for networks of this kind since the network structure is constantly changing. The assignment of a routing protocol is a common issue in these kinds of networks. High-performance routing algorithms must be able to adapt to changes in topology, which happen quickly.

2.3.2.2 Classification based on the Node Configuration

The hardware configuration of the nodes may be used to further classify ad-hoc networks. Configuration is critical, and it may be heavily influenced by the application itself [30]. Homogeneous and heterogeneous node configurations are the two main kinds of networks. A MANET's node.

A. **Homogeneous Ad-hoc Networks:** In homogeneous ad-hoc networks, all the nodes have the same hardware, like a CPU, memory, display, and other parts. Wireless sensor networks are among the best-known examples of homogenous ad-hoc networks. Control components in each node make it easier to locate nodes in homogenous ad-hoc networks; for example, this makes it easier to locate nodes.

B. **Heterogeneous Ad-hoc Network:** Hardware configurations vary across nodes in heterogeneous ad-hoc networks. Each node is unique in terms of its traits, resources, and regulations. 'Node' All nodes in this type of ad-hoc network are unable to deliver the same services

2.3.2.3 Classification based on the Topologies

According to the network topology, ad-hoc networks may be categorised. Flat, hierarchical, and aggregate ad-hoc networks are the three kinds of nodes in the ad-hoc network that have certain tasks [30].

A. **Flat Ad-hoc Networks**: Flat ad-hoc networks have no differentiation between nodes since all nodes have the same responsibilities. All nodes in the ad-hoc network are equal and can perform all tasks. Despite the fact that control messages must be sent throughout the whole network, they are well-suited to networks with highly dynamic topology. Increases in node count have a negative impact on scaling.

B. **Hierarchical Ad-hoc Networks:** This particular form of network is comprised of a number of clusters, each of which stands for a network that is linked to the others. It is possible to classify the nodes that make up hierarchical ad-hoc networks as falling into one of two categories:

- If need to communicate with other clusters, so need a "master node."These nodes are in charge of administering clusters, and they're also responsible for sending data to other clusters.

- A normal node is also referred to as a "slave node." Having no single point of failure is critical to delivering a message. As a result, even if a single node fails, the network as a whole continues to operate as normal. Different rules apply to hierarchical approaches. During the time that a cluster head is down, no messages can be sent or received from other sections of the network to or from that section. For low-mobility applications, hierarchical architectures are more appropriate. Hierarchical architectures are more scalable, but flat architectures are more adaptable and simpler.

C. **Aggregate Ad-hoc Networks:** Zones are formed by aggregating a group of nodes. As a result, the network is divided into many sections called zones. Each node can be part of both a low-level (node-level) and a high-level (zone-level) network topology. Additionally, each node has two ID numbers that may be used to locate it: the node ID and the zone ID. Both of these ID numbers can be utilized. Most of the time, the idea of the zone is linked to aggregate designs. Both intrazone and interzone topologies are available in aggregate structures, which may be flat or hierarchical.

2.3.3 MANET Advantages and Limitations

Although MANET has many advantages, it is due to its inherent nature that it also has many operational limitations. In this section, MANET advantages and Limitations are explained. According to the [31], the main advantages of MANET are:

1. Router Free Connection to the internet without any wireless router is the main advantage of using MANET. Because of this, running MANET can be more affordable than a traditional network.

2. Fault Tolerance; Routing and transmission methods in MANET should handle connection failures.
3. MANET consists of autonomous and mobile parts like laptops, smart phones, wearable computers and tablet computers, PCs and PDAs.
4. Self-organization is possible for the mobile nodes in any temporary network structure.

As a result of its design, MANETs also have several drawbacks [32].

1. MANETs are autonomous and infrastructure-less; hence network administration must be decentralized throughout the network's nodes. This makes it more difficult to find and fix problems.
2. The bandwidth of wireless connectivity is substantially lower than that of wired connections. Wireless LANs now operate at a maximum speed of Mbps, but in wired setups, it is often in Gbps. Due to the restricted capacity of wireless networks, networking tasks, including data routing over several hops, establishing network architecture, and multiple access, have become more difficult.
3. Most mobile nodes in MANETs lack significant processing capabilities. Furthermore, each mobile node has a different software and hardware configuration, which results in different processing power.
4. Constraints on energy: Mobile nodes rely heavily on energy as a resource. Routing, processing, and other tasks need mobile nodes using energy. MANET's mobile nodes cannot run indefinitely since the batteries in these nodes have a limited amount of power. This means that a fundamental restriction of MANETs is a lack of energy.
5. Nodes in a MANET are heterogeneous; therefore, their storage capacity varies. It's possible that certain mobile devices only have a limited amount of memory.

6. A MANET's connection graph is directly impacted by the nodes' ability to move around the network at will. As a consequence of the unpredictability of the network architecture, route breakdowns, packet losses, and network partitions are all too common. Typically, the mobility of nodes has a detrimental impact on the MANET's overall operating capabilities.
7. Instability of connection has a significant impact on a MANET's capacity to perform its intended functions. MANETs have been discovered to have connection issues due to the mobility of nodes, restricted node capabilities, wireless channel impairments, and other issues.
8. In many contemporary MANET applications, such as tactical and sensor networks, hundreds and thousands of mobile nodes are required to operate. For these kinds of networks, scalability is really essential. It is difficult to construct vast networks of nodes with minimal resources.
9. Each node in a MANET has many radio interfaces, each of which may transmit and receive data in a variety of ways. As far as frequencies are concerned, they're all over the map. Since nodes have different radio capabilities, asymmetric connections may result.

2.3.4 Routing Protocols of MANET

Routing protocols for MANETs must be flexible enough to respond quickly to unforeseen topology changes. A successful network management tool must be able to efficiently use the network's resources. Proactive (Table-driven), reactive (on-demand), and hybrid protocols are the three types of protocols [33]. Figure 2.2 illustrates the classification routing protocols of MANET.

In proactive routing protocols, each item in the routing table includes the next hop node utilized in the path toward a destination, regardless of whether the route is now required or not. To keep up with the ever-changing topology

of the network, it is necessary to update the database periodically. Sharing routing information with neighbours in a high-mobility network is a burden for these protocols. However, there will always be routes to locations accessible. Shortest path algorithms are often used by proactive protocols to choose which route to take [34]. DSDV [35], OLSR [36], and WRP [37] are examples of proactive routing.

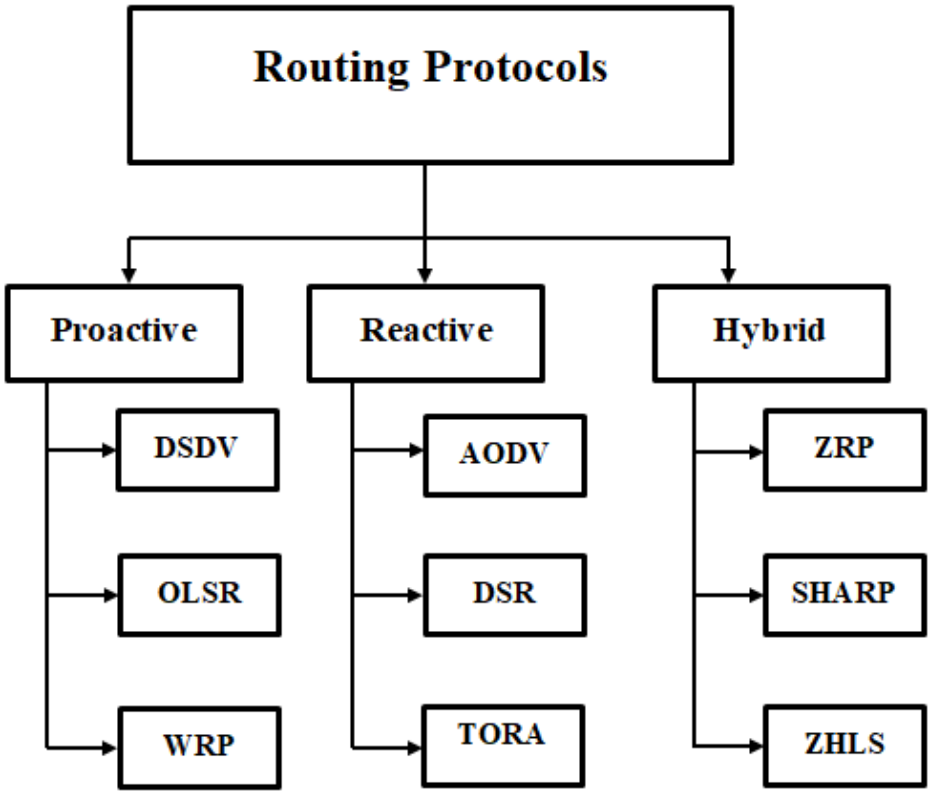

Figure 2.2 Classification of Routing Protocols of MANET

It is un necessary for reactive protocols to keep their routing tables up-to-date with the most recent route topology. For data to be sent, the source node first queries the network for the best path to the destination. Until the target node becomes unreachable or until the route is no longer needed, the route is retained in the database. Cache routes, route discovery mechanisms, and route responses are all handled differently by the protocols in this category. When route discovery is used, reactive methods are regarded to be more efficient. The amount of network traffic created by the route discovery technique is

little compared to the overall communication capacity [38]. Examples of reactive routing are AODV[39], DSR [40], TORA [41] etc.

There are also other types of routing protocols which are reactive and proactive. Both proactive and reactive routing protocols may be used to balance off the disadvantage of table-driven protocols and manage the amount of bandwidth required (in terms of control packages). Proactive routing for short distances and reactive for long distances are hallmarks of the Hybrid Routing protocols [42]. E.g. of hybrid routing are ZRP [43], SHARP [44], ZHLS [45] etc.

2.4 Ad-hoc on Demand Distance Vector

The timer-based status of each node is maintained according to the usage of the routing table in this protocol, which only requests a route when necessary [46]. If a node transmits, accepts, or passes packets to the route, it considers that path to be active. As a result, each node in the intermediate network must make judgments about how to transmit packet data. The network topology changes should only be sent to the nodes that will use them. The issue with Ad-hoc On-demand Distance Vector (AODV)is that it does not allow for symmetrical connections to existing. Asymmetric connections between nodes are supported by this algorithm [47].

Route discovery and route maintenance are two aspects of the AODV routing protocol for MANETs. For discovering and maintaining routes, the AODV protocol uses four sorts of messages: RREQ for Route Request, RREP for Route Reply, RERR for Route Error and Hello for Hello. When a source node wishes to deliver data to a destination node for which there is no record in the routing table, the route discovery procedure is launched [48].

Routing REQuest (RREQ) message is generated when a node wants to communicate with another node that does not belong to its neighbours or has a route to it, or is extremely close to it. The hop count, broadcast ID,

destination IP address, destination sequence number, source IP address, and source sequence number are all included in the Route REQuest (RREQ) message. Additionally, the timestamp is included [49]. When a neighbour node receives a request for an RREQ message, it parses the request to determine the message's destination and then searches its routing database for a path that fits the message's destination.

In AODV, the routing table has the IP address of the destination, the sequence number, the number of hops, the IP address of the next hop, a list of precursors, and the time when the entry will expire. If the node is the destination itself or if it searches for a new route to the destination, it creates and transmits a Route Reply (RREP) message. This message includes the destination IP address, the destination sequence number, the hop count, the source IP address, the lifetime, and the timestamp [50]. AODV route discovery example [39] (see Figure 2.3 and Figure 2.4).

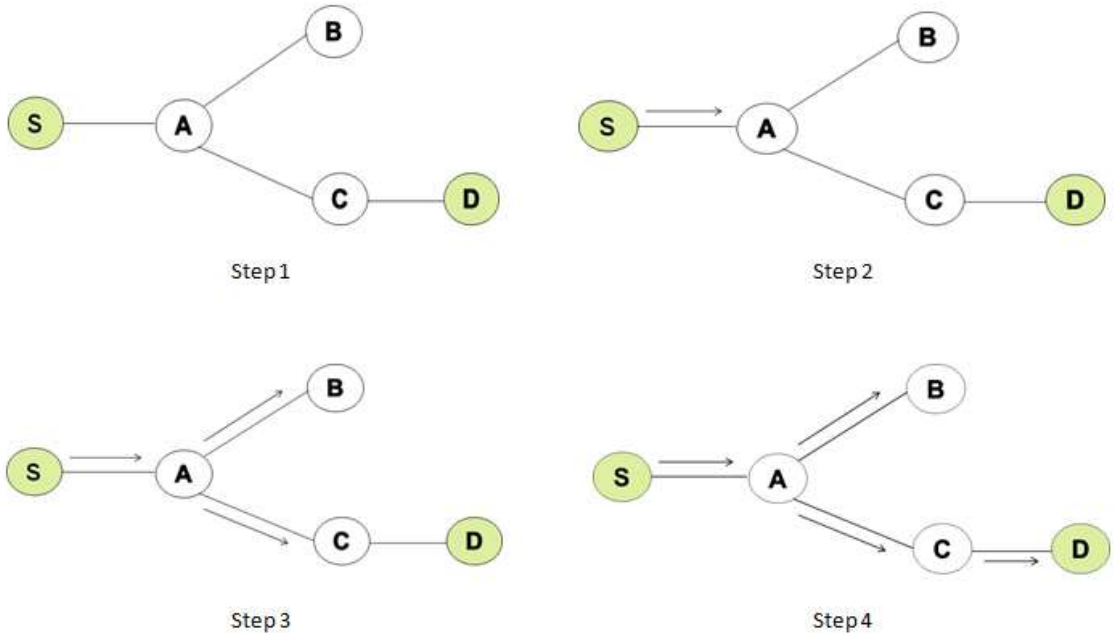

Figure 2.3 Steps of Route Discovery (RREQ)

i. **Route Discovery in AODV**: In Figure 2.3-step1, a route to the destination node ***D*** is requested by the source node ***S***. A RREQ message will have the

IP address and sequence number of nodes ***S*** and ***D***, as well as the number of hops. The RREQ message from node ***S*** will be sent to the nodes around it.

Figure 2.3-step2 shows the RREQ message is received by node ***A***. Nodes ***B*** and ***C*** will receive the RREQ message once again. The RREQ message will be replayed through the reverse route entry to node ***D*** since it has no route there.

Figure 2. 3-Step3 shows how the RREQ message is received by node ***C***. Node ***D*** has a direct path to the source node ***S***, and the reverse route will take it there. When the destination is ***S***, and the next hop is ***A***, the hop count is 2.

Figure 2.3-see Step4 shows that node ***C*** has discovered a path to node ***D*** through nodes ***A*** and ***C***; it will send a RREP message. Thus, the RREP message is broadcast in a unicast fashion back on the same path as the RREQ messages.

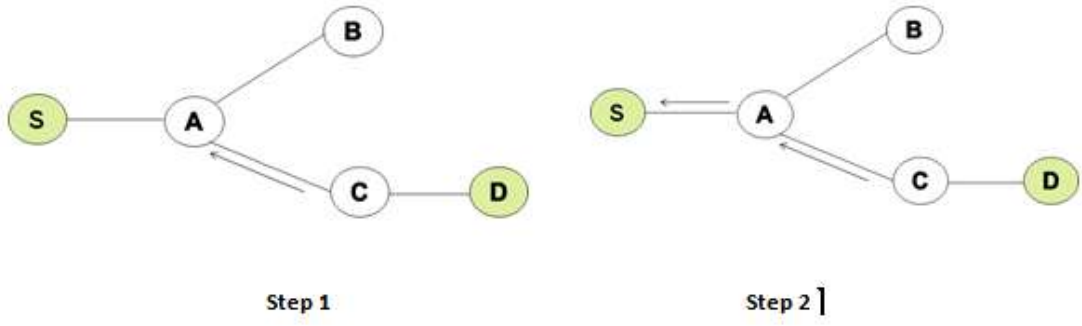

Figure 2.4 Steps of Route Discovery (RREP)

ii. **Route Reply (RREP) in AODV**: It can be seen from the diagram in Figure 2.4-step1, how packets are routed from the source to the destination. When node ***C*** has located the final node ***D***, it notifies neighbour node ***A*** of the route to the final node via node ***C***. When packets are sent from one location to another, Figure 2.4-step2 shows how route

replies are handled. In this case, node ***A*** notifies the source node S about the route to the destination via ***A*** and ***C*** after discovering the destination node ***D*** through node ***C***.

When a path between two nodes is found, the path must be maintained for as long as the source node requires it (Route Maintenance Stage). It is possible for the source node to utilize the route discovery process to identify a new route if it is still in motion. A RERR message is issued to any active nodes that are impacted when the route discovery procedure fails because the source node or any other nodes in the network are moving around. These nodes broadcast the RERR message to their preceding nodes until they reach the source node. Source nodes stop providing data and begin searching for new routes anew when the RERR message is received from a remote node. An intermediate node that has lost connection to its next hop is designated as invalid in the routing database and sends a Route ERRor (RERR) message to all adjacent nodes. Example of Route Maintenance in Aodv [39] (see Figure 2.5):

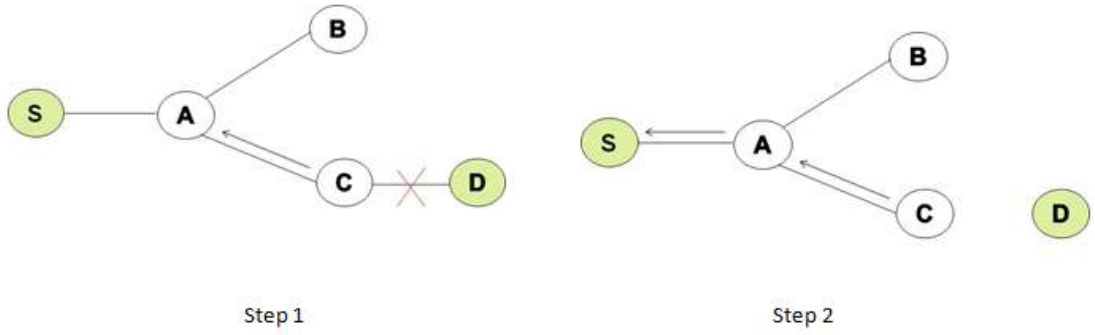

Figure 2.5 Steps of Route Maintenance

iii. **Route Maintenance in AODV**: Figure 2.5-step1 shows that if the connection between node ***C*** and node ***D*** breaks, node ***C*** will send an alarm message to node ***A*** saying that the link to destination node ***D*** has been broken, as seen in the figure. This is done by delivering a RERR message to the nodes that received the packets.

The RERR message that was sent to node ***A*** is resent to the source node ***S*** (see Figure 2.5-step2). This also means that the connection between nodes ***C*** and ***D*** will be deleted since there is no way of getting to node ***D*** from the other nodes. The RERR message is created in AODV whenever any node in the current route fails. It's easy to spot nodes that are not participating in the routing process when they send out the RERR message. This message is sent to adjacent nodes to let them know that a link has failed.

A number of the protocol's characteristics lend themselves to the discovery of selfish nodes (such as the approaches utilized by AODV for this purpose [51][13] [17]). AODV has the following features [39]:

- In order to handle both unicasting and multicasting, the AODV protocol has been developed.
- Routes are built just when requested and with a reduced amount of wait time.
- When AODV is being used, the destination sequence number is utilized in order to locate the most current path to the node that is requested.
- Effectively handle connection failures in the current path.
- Fast reaction to topological changes.
- Excellent for dynamic networks with huge numbers of nodes
- Less time spent setting up the connection.
- Loop-free: AODV.
- No centralized management is required for the routing operation.
- Bidirectional linkages and active pathways from source to destination are reduced in this solution.

2.5 Selfish Node in MANET

Any node in MANET has the ability to display a selfish personality. In order to maximize revenues from network resources, but hesitant to share

its resources with other nodes. In a situation where every node is required to send packets to its neighbors, a few greedy nodes refuse to do that. Except for packets intended for them, these nodes block all traffic. For their own purposes, nodes consume the network and its resources without providing service back. This behavior is characterized by a lack of consideration for the needs of other nodes. As a result, data packets will either be rejected to be retransmitted or will be discarded if they are received by a node that is selfish [17].

Selfish intermediate nodes won't forward packets. Hence, selfish actions will hurt the overall performance of the network as a whole. MANET has more network splitting because there are more nodes that are selfish. One of the biggest problems with MANET is that when the network is split up, the server with the important data is left in a different part. Because of this, it might be harder to find the data [52]. Figure 2. 6 shows the MANET selfish node example.

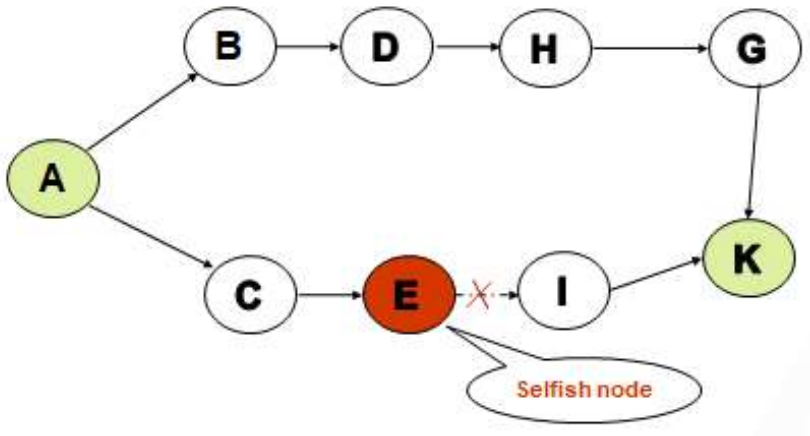

Figure 2.6 Selfish node in MANET

2.5.1 Selfish Node Behaviors

Selfish nodes have a tendency to maximize profits from networks; nevertheless, these nodes are also attempting to preserve resources like bandwidth, battery life, and hardware for themselves. A selfish node will only

communicate with other nodes in the network if its data packet is required to transmit to another node. Additionally, a selfish node will not assist other nodes when it receives data packets or routing packets that it is uninterested in. This behaviour is characterized by a lack of consideration for the needs of other nodes. As a result, data packets will either be rejected to be retransmitted or will be discarded if they are received by a node that is selfish. The selfish nodes' behaviours can be as follows :

- **Nodes Which do not Send Hello Packet**

The principal target of this sort of selfish node is hiding and abstaining from being included in the other transmission way [17].

- **Nodes Which do not Forward RREP Messages**

The whole network will be immobilized as a result of this kind of selfish conduct. To build a transmission line, an RREP message from the destination node to the source node must go through certain intermediate nodes, but these selfish nodes won't send the RREP message. Because of this, the source node will constantly send a RREQ message [53].

- **Nodes Which do not Forward Data Messages**

Nodes like this one that acts in a selfish way hurt the performance of MANET because they drop all the data messages they receive. These packets of data will not be sent out [54].

- **Nodes Forwarding RREQ Messages With Delay**

RREQ messages are sent to this kind of selfish node after a latency approaching the top limit of time out for not participating in a route by this node [53].

- **Nodes Which do not Forward RREQ Messages**

Rather than transmitting these RREQ signals, selfish nodes in MANET simply discard them, preventing them from serving as route members for other nodes in the network. As a consequence, additional nodes are needed to

construct a transmission channel in order to prevent sending these messages to others [53].

- **Selfish Behavior Depending on the Nodes Energy**

Based on the condition, these nodes act in a selfish way. When the energy level is at its highest, it acts like a normal node. When the energy level is between its highest and its threshold t1, it takes part in route discovery and route maintenance, but it does not send the data. When the energy is less than threshold t2, it does not take part in either the phase of finding a route or the phase of sending data [54].

2.5.2 Issues of Selfish node in MANET

The presence of selfish nodes in the network will lead to network partitioning and greatly deteriorating performance. In this section, they explain the issues related to selfish nodes in the network according to Ref. [7]**:**

- **Network partitioning:** The existence of selfish nodes contributes to the increased frequency with which network partitioning happens in MANETs. When the server that has the needed data is separated into a different partition, network partitioning is a significant issue in MANET. This significantly reduces the amount of data that is accessible, which is a significant problem.
- **Data Availability:** If there are selfish nodes in the network, the destruction of certain connections and nodes that are thought to be essential might cause the network to break apart into a number of disconnected parts. Mobile nodes located in one of the partitions are unable to access the data that is being kept by mobile nodes located in the other partition. Because of this condition, there are much less data available.

- **The lifetime of the network:** In a MANET, the performance of the network is greatly reliant on the cooperation of all of the member nodes. A selfish node would often refuse to collaborate in the transmission of packets in order to save its own resources, which may have a significant impact on the lifespan of a network.
- **Throughput:** The availability of selfish nodes in a MANET may have a significant impact on the ratio of the number of packets received by the destination to the total number of packets sent by the source.
- **Hop count:** The portion of the route that is between the nodes of source and destination is referred to as a hop. A hop is represented by one of the nodes along the route that the data takes. In a MANET, the number of intermediary hops between the source and the destination will rise proportionately with the number of selfish nodes. It's possible that the Network's performance may suffer as a result.
- **Packet dropping Ratio:** The number of packets that are ignored by routers because certain nodes are acting in a selfish manner in order to save their resources.
- **Packet Delivery Ratio:** This refers to the proportion of the total number of data packets that have been sent from the source node to the destination node. It is because of a selfish node in the MANET that it gets impacted.
- **End-to-End delay:** The end-to-end delay refers to the amount of time it takes for a data packet to be sent from a source node to a destination node while travelling across a MANET. It is boosted by MANET nodes that are selfish.
- **Probability of Reachability:** The fraction of feasible reachable paths that can be taken to any and all conceivable routes between any and all source and any and all destinations.

2.5.3 Selfish Node Detection Techniques

In MANET, there are several ways to identify a selfish node. In the following, explain of the techniques based on credit, reputation, or acknowledgement.

- **Credit-Based Technique:** The primary goal of the credit-based technique is to influence the network nodes to perform at their optimal levels. Plastic money has been introduced as a means of achieving this goal. Every node is rewarded for providing services to its neighbours in this manner. When a node asks a neighbour node to forward a packet, they compensate the neighbour node using the same virtual money mechanism [54]. Examples of methods to detect selfishness that used this technique in [55][56][16].
- **Reputation-Based Technique:** As each node monitors the transfer of a packet to another node or obtains information about other nodes from a central node, a reputation-based method is used. The reputation of a node is boosted or diminished depending on whether or not it effectively contributes to the transport of data by forwarding data packets. Nodes are penalised or ignored if their reputation falls below a certain level defined by the developer [7]. Examples of methods to detect selfishness that used this technique in [57][15] [17].
- **Acknowledgement-Based Technique:** To determine whether or not a packet has been transmitted, an ACK is required [58]. It has a number of Benefits. The difficulties that are caused by the other techniques, such as ambiguous collisions, receiver collisions, and restricted transmission power, may be solved by using the ACK. Despite the fact that ACK has a number of drawbacks, When compared to other techniques, the ACK technique has a much larger routing overhead[59].

Examples of methods to detect selfishness that used this technique in [60][13].

2.6 Simulation Environment

When it comes to designing and simulating networks, network simulators provide an integrated, flexible, and user-friendly solution for based network designers. It is suitable to implement the network in the actual world using a variety of various tools that are designed specifically for network simulation. This section featured simulators and tools that are used in this work.

2.6.1 Network Simulator-2

Network Simulator-2 (NS-2) is a network simulator that is available for free. NS-2's open-source nature and extensive component library have led to its widespread adoption in the academic community. The real network simulator serves as the foundation for NS. Because of the large number of packages that have been provided to NS-2 by a variety of nonprofit organisations, it is one of the most popular tools for simulating network behaviour. A network simulator that is object-oriented and controlled by discrete events, NS-2 is known as The programming languages C++ and OTcl are used in NS-2.

NS-2 divides control path implementations from data path implementations into two distinct categories. In order to speed up the process of handling packets and events, the event scheduler, as well as the fundamental network component objects in the data path, have been created and built in C++. Users may manage the simulation scenario and schedule events via the use of OTcl, while C++ is utilised to construct the detailed protocol. The event scheduler is another function that may be performed using NS-2. The event scheduler in NS-2 is responsible for monitoring the passage

of simulation time and releasing all of the events that are stored in the event queue by activating the relevant network components [61].

Noted that the main reason for using NS-2 to test and analyze the proposed approach is that most related works use it to detect selfish nodes, as shown in Table 1.1 in the first chapter.

Other simulators that are generally acknowledged as being the most common ones used for studies regarding selfish nodes in MANETs are Network Simulator-3 [62], QualNet [63], GloMoSim [65].

2.6.2 Tool Command Language

A Tcl, which stands for "Tool command language," is a robust scripting language that also includes programming elements. It is downloadable for use on systems running Unix, Windows, and Mac OS. Tcl is used for graphical user interfaces, Web and desktop applications, networking, administration, testing, and fast prototyping, as well as scripted application development. Need to utilize a Tcl in order to configure the simulation network in NS-2. In reality, it makes use of an extension of Tcl known as OTcl (object-oriented extension of TCL), which adds support for objects to standard Tcl. In NS-2, a configuration file is an OTCL file that is referred to as a "TCL Simulation script." It also provides information on the things that would want to replicate, such as the development of nodes and topologies, the establishment of links, and other similar things that may use a TCL file as an input configuration file for a C++ file as well. The TCL file is used as an input configuration file for the ns file, which is an executable file [66].

2.6.3 AWK

The text scanning and manipulation method was developed by Aho in collaboration with Weinberger and Kernighan (hence the label). Each line of the input file is processed by the awk software in turn. After that, there is the

main section that operates on each line of the file, and finally, there is an END section that contains the operations that take place after the file has finished being read. It is possible for it to include an optional BEGIN section of commands that are processed before any content is processed from the file. If there are any instructions that match the pattern, then it works only on lines that match this pattern. If there are no instructions that match the pattern, then it works on all of the fonts. This check is performed on each line of the input file. AWK commands are capable of doing some highly complicated mathematical computations and string manipulation, and they also support AWK matrices [67].

2.7 Performance Metrics

Packet Delivery Ratio, Throughput, Average End to End delay, Packet Retransmission and Power Consumption Represent are the performance metrics that have been used in this work.

- **Packet Delivery Ratio**

The ratio of the number of packets received by the destination to the number of packets created by the source node is referred to as the Packet Delivery Ratio (PDR) [68]. Equation (2.1) is used to calculate PDR.

$$\mathbf{PDR} = \left(\frac{\mathbf{No.of\ the\ packet\ received}}{\mathbf{No.of\ the\ packet\ sent}}\right) * \mathbf{100}\% \quad \ldots\ldots\ldots (2.1)$$

- **Throughput**

It is one of the dimensional metrics of the network that determines the amount of the channel capacity that is really being utilized for productive transmission. Chooses a destination at the start of the simulation; this provides information on whether or not data packets were successfully

delivered to their respective destinations [69]. Equation (2.2) is used to calculate Throughput.

$$\textbf{Throughput(kbps)} = (\frac{\textbf{output data (byte)}}{\textbf{Times}}) * (\frac{\textbf{8}}{\textbf{1024}}) \quad \ldots\ldots\ldots (2.2)$$

Where output data (byte) represents the quantity of data that was delivered by the sender, and *Times* refers to the length of time it took for the data to reach its destination.

- **Average End-to-End Delay**

End-to-end packet delay may be described as the difference in time between the time moment at which the packet reaches the receiver and the time instant at which the packet is formed at the sender. This difference in time is called the end-to-end packet delay. It determines how much time passes between the packet's departure from the sender and its arrival at the application being used by the recipient. It is possible for there to be delayed as a result of the queue or medium access control, as well as the propagation of the channel [70]. This should be a low metric. Equation (2.3) is used to calculate Average End to End Delay (AE2E).

$$\textbf{AE2E} = \frac{\textbf{sum of the time spent to deliver packets for each destination}}{\textbf{number of packets received by the all destination nodes}} \quad \ldots. (2.3)$$

- **Packets Retransmission Rate**

Resending data packets that could not be sent successfully around the first time due to corruption or loss is what is meant by the term "packet retransmission" [71]. Equation (2.4) is used to calculate Packets Retransmission Rate (PPR).

$$\textbf{PPR} = \frac{\textbf{No.of packets sent} - \textbf{No.of the packet received}}{\textbf{No.of packet sent}} * \textbf{100} \quad \ldots\ldots (2.4)$$

- **Power Consumption**

The term " power consumption " refers to the amount of energy that is consumed by each node[72]. Equation (2.5) is used to calculate power consumption.

$$\textbf{Power Consumption} = \textbf{Initialenergy} - \textbf{Remaining Energy} \ldots..(2.5)$$

CHAPTER THREE

RESEARCH METHODOLOGY

AND

PROPOSED METHOD

3.1 Introduction

This chapter provides a review of the research methodologies utilized throughout the thesis. The research design used for the goals of this study is described, along with the reasoning behind this selection. Block diagrams and algorithms are also included to illustrate the proposed method functionality. This chapter represents a practical partition of this work. The chapter starts with the overall research methodology framework, as shown in the next section. Section 3.3 shows the test and analysis scenario using the NS-2 environment. The conceptual and analytical model are demonstrated in Section 3.4. Section 3.5 describes Least Energy and Least Communication Ratio (LELCR) scheme. Section 3.6 presents the Isolate Selfish Node scheme. Detection, Reintroduced, and Collaborative of Selfish Node (DRSCN) scheme is proposed in Section 3.7. Finally, the evaluation of the proposed method is presented in Section 3.8.

3.2 Research Methodology

The performance of MANET is dependent upon the cooperation of all nodes in the network for path discovery, path maintenance, and forwarding packets to each other. Selfish behavior of nodes degrades network performance, and active nodes become unfairly overloaded. The research methodology of this work consists of a number of stages that complement the work together to implement the proposed method. Figure 3.1 shows the proposed method block diagram, which shows the macro view structure for the general stages taken to achieve and evaluate the proposed method. It includes four stages.

In the first stage of method modeling, where analyze the related theories for the detection of selfish nodes and suggest an method to improve the selfish node instead of isolating it from network activities. Detection, Reintroduced, and Collaborative of Selfish Node (DRCSN) scheme is the

main contribution to this study. Besides this scheme, the proposed method has two schemes: a special scheme for detecting selfish nodes based on Energy and Communication Ratio (LELCR) and the second scheme to isolate selfish nodes. Also, building the required algorithms based on these three schemes. The second stage is the experimental design of the proposed method via choosing a suitable language, which is represented here (c and tcl), converting the schemes from algorithms to programming code, choosing a simulation program NS-2, and determining the appropriate parameters. The third stage is verification and validation, which include implementation verification, design validation, and ensuring method confirmation. And finally, the fourth stage evaluates the proposed method by evaluating performance metrics and comparing it with related works.

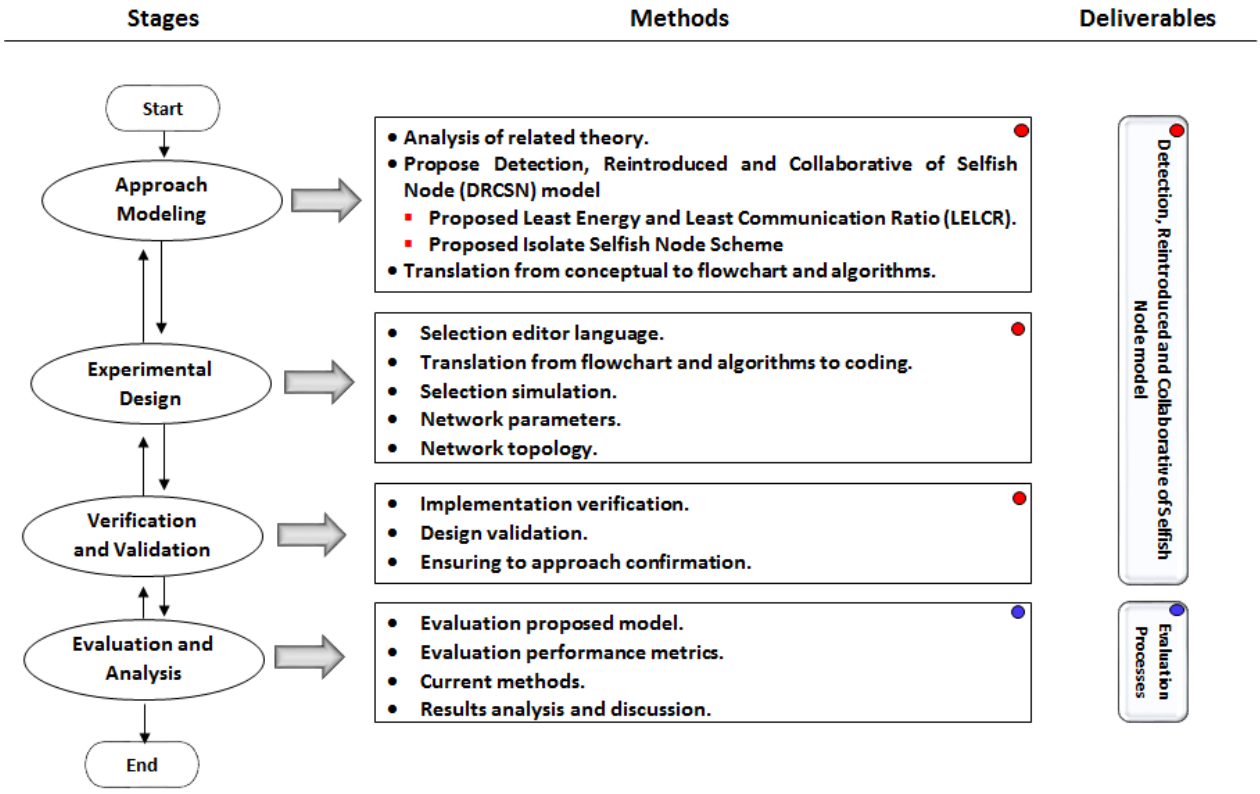

Figure 3.1 Research Methodology and Macro View for the Proposed Method

While Figure 3.2 shows the proposed Method block diagram, which shows the micro view structure with deep details of the proposed method.

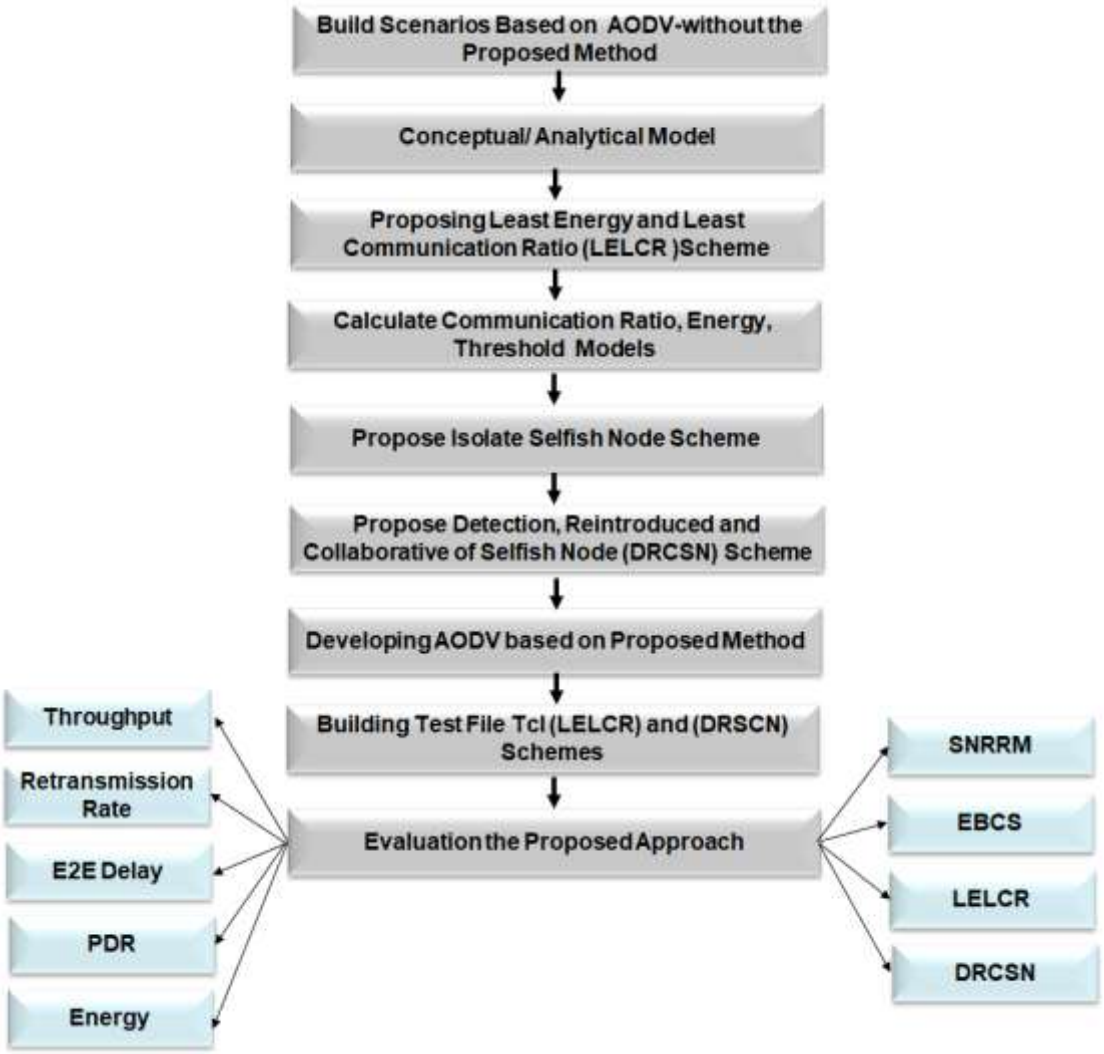

Figure 3.2 Research Methodology and Micro View for the Proposal Method

The research methodology of this work consists of a number of steps that complement the work together to implement the proposed method. The first step includes testing the scenario based on the NS-2 environment under different numbers of nodes and speed nodes, as well as analysis of the impact of selfish nodes on both the energy and the communication ratio (Section 3.3). After that, the conceptual/ analytical model is explained (Section 3.4). Then, proposing a Least Energy and Least Communication Ratio (LELCR) scheme, which includes building three models: energy, communication ratio, and threshold of energy, it will be explained in detail in (Section 3.5). Next, this paper will explain the scheme responsible for isolating the selfish nodes (Section 3.6). Then, it will explain how to make selfish nodes collaborative by

reducing the rate through the proposed scheme Detection, Reintroduced, and Collaborative of Selfish Node (DRCSN), which represents the main objective of this study (Section 3.7). The last step represents the evaluation part of this work by comparing DRCSN with each of (SNRRM, EBCS, and LELCR) (Section 3.8).

3.3. Test and Analysis Scenario Using NS-2 Environment

This section includes conducting an NS-2 test according to sequentially changing the network parameters (number of nodes and speed of nodes). MANET scenarios are created according to the simulation parameters that will be defined (as shown in Table 4.1), and a test is conducted in both cases (the first case when the number of nodes is variable from 20 to 100 and the speed of node is fixed 10 m/s, and the second case when the number of nodes is fixed 100 and the speed of node is variable from 5 to 25 m/s.

The performance will be evaluated based on five metrics (packet delivery ratio, packet retransmission rate, throughput, average E2E delay, and power consumption). The purpose of this stage is to analyze the network performance in both the impact of the number of nodes in the network when it is few and large; and the impact of nodes speed when it is slow and fast speed.

3.4 Conceptual/ Analytical Model

Previous works focused on isolating selfish nodes from network activities only as the main tool to address the problem of selfish nodes in the network. The main objective of this research is to improve network performance by exploiting selfish nodes and making them cooperate to the maximum extent. Method modelling is the main stage in this thesis because it includes the conceptual model. The proposed method consists of the basic principles: Detection, Isolation, and Collaboration. Where the detection represents a scheme to detect selfish nodes based on the communication ratio

and energy by (Least Energy and Least Communication Ratio LELCR Scheme), this scheme needs another scheme to isolate the selfish nodes from the routing table, it represented by (Isolate Selfish Node Scheme). Finally, collaboration is represented by making the selfish nodes collaborative and exploiting them to the maximum extent by reducing the rate of a packet for selfish nodes after detection; it is represented by (Detection, Reintroduced and Collaborative of Selfish Node DRCSN Scheme). The proposed method is the result of merging all these schemes, where the effect on network performance is positive in terms of throughput, residual energy, and PDR, as well as a positive impact in terms of packet loss and delay. Hence, network performance has become better. As shown in Figure 3.3, the conceptual modelling and basics of the proposed method are illustrated.

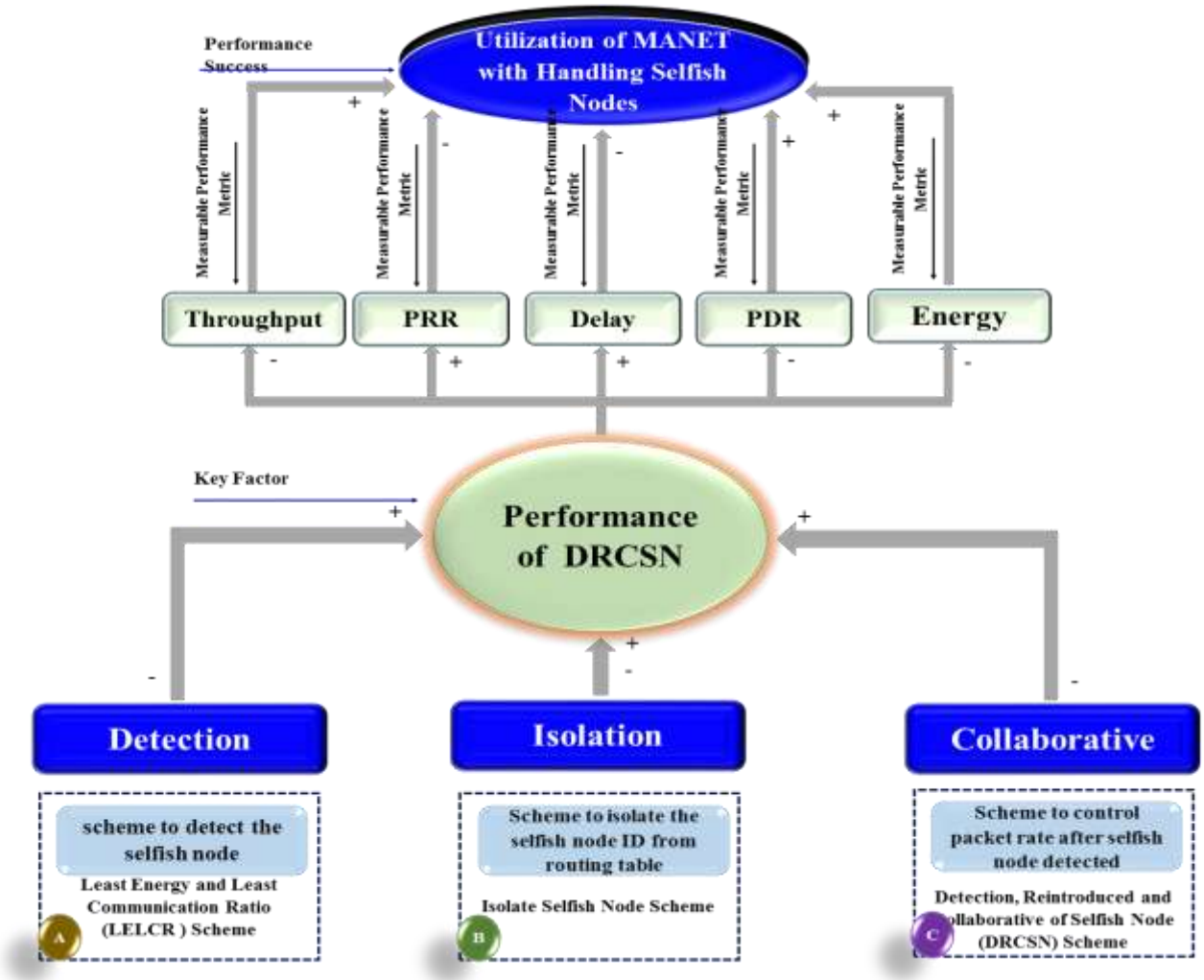

Figure 3.3 Proposed Conceptual Model

3.5 Least Energy and Least Communication Ratio Scheme

In MANET, nodes are self-configured for packet transfer from one site to the next. Within a MANET, the behaviour of the nodes might take on a variety of types, whereas selfish nodes will have an effect on the whole network. A selfish node tries to get all of the resources for itself and does not want to share them with its neighbours. So, to make MANET work better, it is very important to find the selfish nodes.

There are different types of selfish nodes. Adopt here selfish nodes that do not forward RREP messages as well as selfish nodes that depend on energy. Depending on the Communication Ratio (CR) and Energy, this thesis managed to detect selfish nodes in the MANET because it has a significant and subtle impact on selfish node detection. The following section detail is calculating Energy and CR in detecting selfish nodes.

3.5.1 Calculate Communication Ratio

The node's activity is monitored based on reply messages received from other nodes in the network. The node CR is established based on the nodes' behavior. If the value of CR is greater than 30%, the node is normal and capable of sending and receiving data. If the value of CR is less than 30%, then the node is considered selfish.

The CR is computed based on the route request and route reply messages transmitted throughout the communication network. The CR is computed for a network node based on the disparity between the number of route request messages that have been received and the number of route reply messages that have not yet been sent in comparison to the total number of route request messages that have been received. The CR for each node is calculated according to Equation 3.1 and Equation 3.2:

$$unsentMassegenode = GRRnode - SRRnode \quad \ldots\ldots\ldots (3.1)$$

$$CR = (GRR - unsentMassegenode / GRR) * 100 \quad \ldots\ldots\ldots\ldots (3.2)$$

Where:

GRR: Get Route Request.

SRR: SendS Route Reply.

3.5.2 Calculate Energy

In MANET, data distribution is highly dependent on the collaboration of all nodes with each other, and sometimes nodes will not forward packets to their neighbours in order to save as much energy as possible to use for their own benefit. This selfishness prevents nodes from collaborating with each other, which affects the performance of the network. In order to detect the selfish nodes in the network, the paper relied on the residual energy as follows (Equation 3.3):

remainingEnergy = Initial Energy – Consumed Energy …… (3.3)

Where residualEnergynode represents the current energy of the nodes.

In the next section, the threshold equation that is used with Energy in the detection of selfish nodes is clarified.

3.5.3 Calculate Threshold of Energy

After calculating the Energy for each node included in the network, the following is the threshold equation (Equation 3.4) that is used to detect the selfish nodes based on residual energy during the simulation time:

Threshold=((IEnode– remainingEnergy/residualEnergy)*currentTime)... (3.4)

Where:

IE: Initial Energy

Threshold: Threshold of energy

If the remaining energy in the node becomes less than the threshold, this node will be identified as a selfish node.

3.5.4 Algorithm of LELCR

In this section, the algorithm (LELCR) is developed based on the equations CR, energy, and threshold of energy. The reputation of each node is calculated by the current energy level of the node and its CR. In the next stage, the results will be analyzed at each node in order to determine the value of both the node CR and the energy value in a given time period. This identifies nodes that may be responsible for selfishness.

Before the source node starts to send packets to a destination node, it lookups at its routing table for a routing route that leads to the destination node, the source node will send an overwhelming number of route request RREQ packets to its neighbors in the event that the route to the destination node cannot be located. The available threshold, residual energy, and CR are the three fields that are included in the RREQ packet that is part of the AODV protocol. According to the RREQ packet that AODV uses. If the available energy is lower than the energy threshold and the CR is lower than 30% (where a value of 30% has been imposed depending on [73]), then the mobile nodes in question are considered to be selfish.

Relying on Equations (3.1),(3.2),(3.3),(3.4), the previous parts, suppose that GRR = 80, SRR = 20

unsentMassege = 80 - 20 = 60

CR =((80 - 60) / 80) * 100 = 25

As for the second equation, Assuming that IE = 100, and the consumed energy = 40 per second 40 in the following:

remainingEnergy = Initial Energy – Consumed Energy

remainingEnergy = 100 – 40

residualEnergy = 60

Threshold = (IE - remainingEnergy / remainingEnergy) * current Time

Threshold = ((100 – 60) / 60)* 40= 26.666

Since the CR is less than 30 and the residual energy is less than the threshold 26.666 then the node is considered selfish, in this case, isolating the selfish nodes is done from the network activities.

Algorithm 3.1 is presented the proposed Least Energy and Least Communication Ratio scheme.

Algorithm 3.1: Least Energy and Least Communication Ratio

Definitions:

nNodes : Number of nodes
GRR: Get Route Request
SRR: Send Route Reply
IE: Initial Energy
CE: Consumed Energy
CR: Communication Ratio
remainingEnergy: Residual Energy
Threshold: Threshold of Energy

Begin:

1. For each $node_i \in$ nNodes do
2. Threshold $\leftarrow 0$
3. Calculate the remainingEnergy$\leftarrow$ IE_node_i - CE_node_i
4. Calculate the unsentMassege$node_i \leftarrow$ GRR_node_i – SRR_node_i
5. Calculate the $CR\_node_i \leftarrow ((GRR\_node_i - unsentMassege\_node_i) / GRR\_node_i) * 100$
6. Threshold $\leftarrow ((IE\_node_i - remainingEnergy\_node_i) / remainingEnergy\_node_i) * currentTime$
7. if $CR\_node_i < 30\%$ and $remainingEnergy\_node_i <$ Threshold then
8. Call isolateSelfishNodes ($node_i$)
9. Otherwise, $node_i$ is cooperative
10. End_if
11. End_for

End_Algorithm

3.6 Isolate Selfish Node Scheme

After defining the nodes to be isolated, these nodes are isolated through the Selfish Isolate scheme, which isolates the selfish nodes from the routing table. When the selfish node is detected, it will be deleted from AODV routing table, thus reducing the number of contents of the routing table to

minus one. When resending the Hello messages, if the routing table is not full, a new ID will be entered for it, provided that ID is not present in the routing table or is back to a selfish node that was previously deleted. Algorithm 3.2 is presented the steps involved in isolating selfish nodes.

Algorithm 3.2: Isolate Selfish Node

Input :
$node_i$: Selfish node

Output :
Delete the route of the selfish node from $node_i$ routing table

Begin

1. recordRouting Table Lookup ($IDnode_i$)
2. if recordRouting Table ! = 0 then
3. AODV_Delete ($IDnode_i$)
4. AODV_routing Table - 1
5. End_if
6. AODV broadcast Hello message
7. if AODV_routing Table is not full then
8. if new IDnode is not $IDnode_i$ and new IDnode is not included in AODV_routing Table then
9. AODV_Add (new IDnode)
10. AODV_routing Table + 1
11. End_if
12. End_if

End Algorithm

An example (see Figure 3.4 and Figure 3.5) of the process of isolating the selfish nodes. We printed a rotating table of node 40 before and after isolation and executed it on 100 nodes, and the result in.

Where the first column represents the "node:", the second column represents node id, the third column represents current time, the fourth column is represents the destination, the fifth column represents next hop, while the sixth, seventh, eighth and ninth columns represent hops, sequence number, expire, flags, respectively.

```
NODE: 40 5.008201 78 78 1 42 1 15.008201 -261804494
NODE: 40 7.075384 80 46 5 26 1 17.075384 1706478502
NODE: 40 7.075384 76 26 4 42 1 12.026586 -1621875971
NODE: 40 7.075384 50 26 7 36 1 11.054306 54401946
```

Figure 3.4 Example before Isolating the Selfish Nodes

```
NODE: 40 5.008201 78 78 1 42 1 15.008201 -261804494
NODE: 40 7.837796 80 18 5 28 1 17.837796 -871814150
NODE: 40 7.837796 76 9 3 44 1 12.773902 -750320876
NODE: 40 7.837796 50 26 7 36 1 11.054306 54401946
NODE: 40 8.091804 30 59 4 28 1 13.774059 -2085404760
NODE: 40 8.091804 23 29 2 32 1 18.091804 1882722546
NODE: 40 8.091804 50 33 2 36 1 11.017492 -1071945630
```

Figure 3.5 Example after Isolating the Selfish Nodes

3.7 Detection, Reintroduced and Collaborative of Selfish Node Scheme

All existing methods of manipulating selfish nodes merely detect and isolate them from network activities. In this study, a basic scheme (LELCR) was developed to deal with selfish nodes and exploit them to their fullest instead of isolating them. This scheme is an extension of all the schemes mentioned above and adds the possibility of making selfish nodes have to cooperate by controlling the packet rate, as long as nodes have enough power to send and receive packets. It will give them packets they can handle. Thus, reducing the burden on selfish nodes to remain efficient in the network and not cause damage to the network. Thus, the proposed method achieved its purpose and it will lead greatly improve network performance.

After detection of selfish nodes depending on the residual energy and CR, when the residual energy in the node is between threshold and 10% (where this thesis assumed that the energy needed by the nodes to be effective in the network is 10% of its initial energy) and CR is less than 30%, it will

classify the nodes as selfish but exploitable. In this case, it will be reduced the number of requests sent to it in order to be employed as much as possible. But, when the energy of the nodes is less than 10%, the isolation scheme will be called to isolate it from network activities and replace it with another node. Algorithm 3.3 is presented as Proposed Detection, Reintroduced and Collaborative of Selfish Node Scheme.

Algorithm 3.3: Detection, Reintroduced & Collaborative of Selfish Node

Definitions:

nNodes : Number of nodes
GRR: Get Route Request
SRR: Send Route Reply
IE: Initial Energy
CE: Consumed Energy
CR: Communication Ratio
remainingEnergy: Current Energy
Threshold: Threshold of Energy

Begin:

1. For each $node_i \in nNodes$ do
2. Threshold ← 0
3. Calculate the $remainingEnergy\_node_i \leftarrow IE\_node_i - CE\_node_i$
4. Calculate the $unsentMassegenode_i \leftarrow GRR\_node_i – SRR\_node_i$
5. Calculate the $CR\_node_i \leftarrow ((GRR\_node_i – unsentMassege\_node_i) / GRR\_node_i) * 100$
6. Threshold ← $((IE\_node_i – remainingEnergy\_node_i) / remainingEnergy\_node_i) * currentTime$
7. if $CR\_node_i < 30\%$ and $remainingEnergy\_node_i <$ Threshold then
8. if $remainingEnergy\_node_i > 10$ then
9. Update Rate ← Rate / 2
10. Otherwise, Call isolateSelfishNodes ($node_i$)
11. End_if
12. Otherwise, $node_i$ is cooperative
13. End_if
14. End_for

End_Algorithm

Figure 3.6 is an example that shows the work of the AODV protocol in the NS-2 environment with the presence of selfish nodes in the network. Let's

assume that Figure 3.6 (a) has a MANET network consists of 9 nodes; node ***A*** represents the source, while node ***K*** represents the destination.

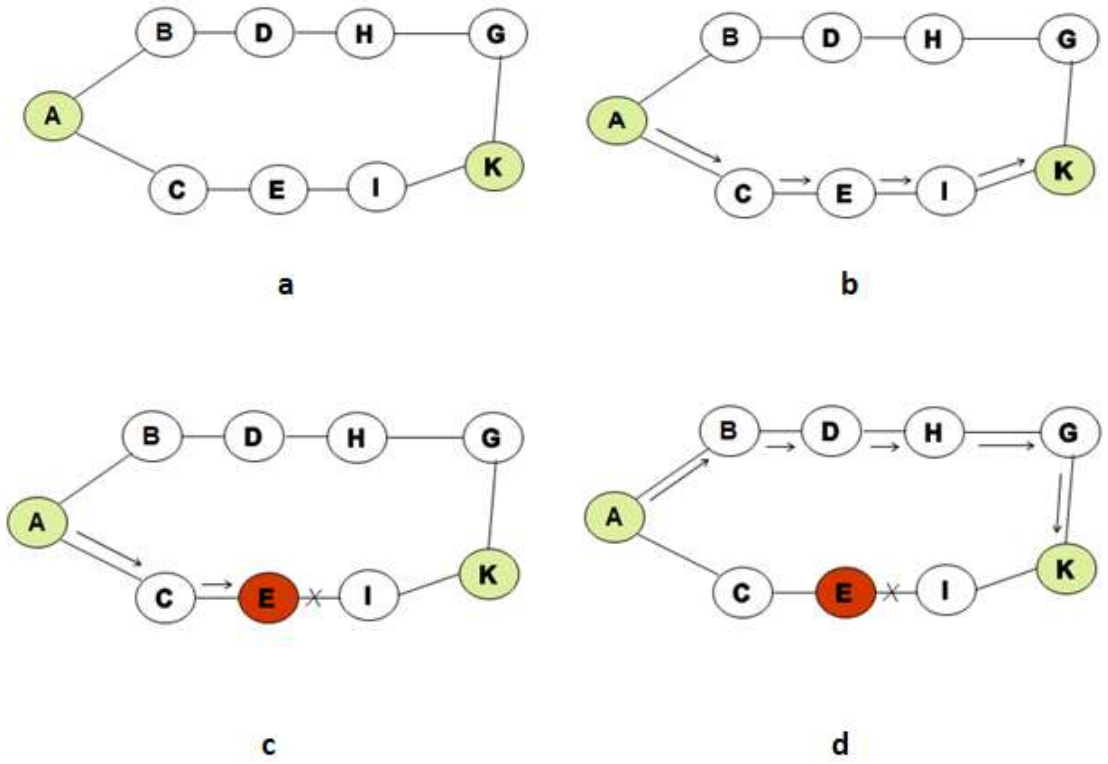

Figure 3.6 Example of AODV Protocol

After the route discovery process via (Hello messages), (RREQ messages) and (RREP messages), the path as shown in Figure 3.6 (b) has been chosen to reach the source. During the packet transmission process, each node will watch its neighbours. This is done to determine whether or not the nodes that are close are active. All nodes forward packets from source to destination. In some cases, not all of the nodes in the network will function perfectly.

Also, let's assume that node ***E*** is selfish; the path between node ***E*** and ***I*** will fail, as shown in Figure 3.6 (c). In this case, source node ***A*** receives a message (RERR) telling it there was a failure to reach the destination node ***K***. RERR message identifies nodes that do not contribute to routing. This message is issued to inform the nodes that are next to the failed connection that it has occurred. Then, the source node will choose an alternate second path, as shown in Figure 3.6 (d).

In the next example, when the development of AODV is based on the DRCSN scheme. After detecting the selfish node specified as ***E***, node ***C*** will reduce the number of requests sent to node ***E***, thus optimizing the selfish node and completing its path with the best path to destination ***K***. As shown in Figure 3.7.

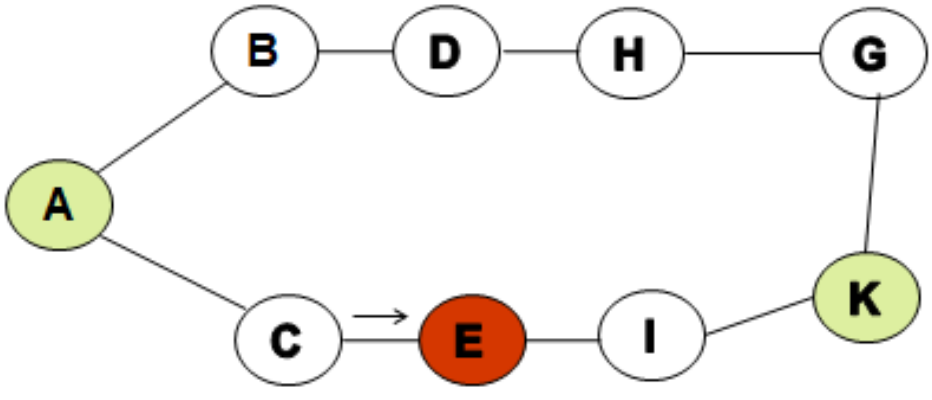

Figure 3.7 Example of AODV Protocol based on DRCSN Scheme

Then developing AODV based on proposed method and building Tcl file for testing LELCR and DRSCN Schemes (details shown in the appendix).

3.8 Evaluation of the Proposed Method

In this section, evaluate and analyze the results to ensure that the method is correct and that it reaches its goal. Five scales are used in this thesis (details in 2.7), including Packets Retransmission Rate, Packet Delivery Ratio, Throughput, End to End Delay, and Power Consumption. The results of these metrics are obtained from reading files produced by awk which include all the details for each scenario.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

Through the introduction chapter, which covered the general overview, related works, problem definition, the research objectives, and the research scope. While Chapters Two provided an overview of the background of this research. It was made clear to explain and criticize the related works which were used to solve the problem. Chapter Three sets up the research methodology as a guide for achieving the goals of this research. It also showed all the steps needed to evaluate the developed method's performance.

In this section, the paper will discuss the results of the study, analyze quanitative data. The results are also addressed in the context of findings via implementing previous research and the accessible literature in order to highlight similarities and contrasts between the findings of this study and those of previous studies and literature. This chapter is organized as follows. Section 4.2 explains the simulation setup. The validation and evaluation of the proposed method, performance evaluation without the proposed method, and performance evaluating of DRCSN and LELCR in Sections 4.3, 4.4, and 4.5, respectively. Finally, Section 4.6 performance evaluation of DRCSN, SNRRM, and EBCS.

4. 2 Simulation Setup

NS-2.35 is used extensively for simulation. The simulated network includes 20–100 mobile nodes spread out over an area of 1000x1000 square meters. The mobile speed nodes are set from 5–25 m/s. Each packet included 512 bytes long.

Additionally, it is assumed that each mobile node has 100 joules of energy and that each communication time slot needs 10 joules of energy. The simulation settings that were configured for our research are outlined in Table 4.1.

Table 4.1 Simulation Parameters

Parameters	Values
Simulator	NS-2.35
No. of Mobile Nodes	20, 40, 60, 80, and 100
Terrain area	1000mx1000m
Mac Layer	Mac/802.11
Interface Queue Type	Queue/DropTail/PriQueue/MUP
Mobility Model	Random Waypoint
Traffic Source	CBR (4.0 packets/sec)
Packet Size	512 Bytes
Protocol	AODV, DRCSN
Propagation Type	Two Ray Ground
Antenna Model	Antenna/OmmniAntenna
Every mobile node contains energy	100 joules of energy
Required for each time slot of communication	10 joules of energy
Net Interface Type	100 Phy /Wireless Phy
Max Packet in IFQ	50
Speed of Nodes	5, 10, 15, 20 and 25 m/s
Max Number of Connections	3
Simulation Time	100s

4.3 Validation and Evaluation of the Proposed Method

The proposed method is validated and evaluated by implementing it in two scenarios (changing the number of nodes and changing the speed of

nodes). In addition, the evaluation process is done through five performance evaluation metrics (throughput, power consumption, packet delivery ratio, packets retransmission rate, and average E2E delay) and comparing the obtained results with the current related works. Finally, check whether the proposed method fulfill its purpose or not.

4.4 Performance Evaluation without the Proposed Method

In this section, the test and analysis scenarios are implemented using the NS-2 simulator, as it is mentioned in Chapter Three. For this purpose, many scenarios are created according to two important network parameters, which are the number of nodes, and speed of nodes. In the first scenario, the number of nodes is set from 20 to 100, and the speed node is fixed as 10m/s. In the second scenario, the speed of node is setting from 5 to 25 m/s, and the number of nodes is fixed as 100.

4.5 Performance Evaluation of DRCSN, LELCR and AODV-without the Proposed Method

In this section, the DRCSN and LELCR schemes scenarios are implemented and tested within the NS-2 simulator. DRCSN and LELCR schemes are implemented inside the AODV protocol to develop a new version of the MANET protocol and test it under many scenarios. In order to evaluate how well the proposed schemes perform under different circumstances, two categories of situations have been taken into consideration (impact of a number of nodes and impact of a variety of speed of nodes).

4.5.1 Impact of the Number of Nodes

In order to conduct an investigation on the influence that the number of nodes has, the maximum number of nodes was changed from 20 to 40 to 60 to 80 to 100, while the speed node remained at a constant 10 meters per second. Figures 4.1 through 4.5 illustrate the influence that the number of nodes has

on the different performance parameters for DRCSN and LELCR schemes, respectively.

❖ **Throughput**

Figure 4.1 presents a comparison between the DRCSN, LELCR and AODV-without the Proposed Method in terms of their levels of performance with regard to throughput. In the illustration that has been shown, the values of the throughput are indicated along the y-axis, whilst the number of nodes is shown along the x-axis.

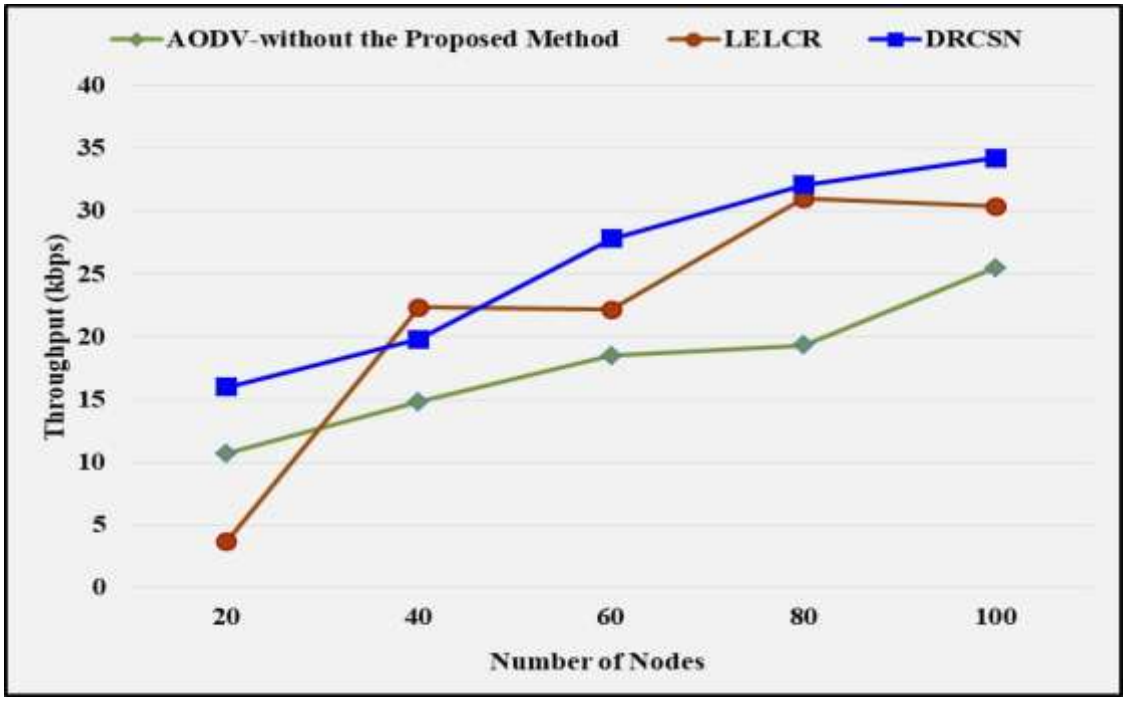

Figure 4.1 Impact of Number of Nodes vs Throughput for DRCSN, LELCR and AODV-Without the Proposed Method

As shown in above figure, when the number of nodes goes from 20 to 100, the throughput of the DRCSN scheme will be better than LELCR scheme, except for only one case when the number of nodes is 40, the throughput of LELCR scheme will be better, where the throughput of the DRCSN is 19.79034 kbps, while the values of LELCR is 22.31798 kbps. The main reason for that, can be assigned to the nature of the architecture and the basic

principles of networks themselves. Since the packets are impacted by devices which are natural and the fluctuations of the intermediate network.

❖ Packets Retransmission Rate

As noted in Figure 4.2 comparison between the DRCSN, LELCR schemes and AODV-without the proposed method in terms of their levels of performance with regard to packets retransmission rate. Although DRCSN, LELCR schemes and AODV-without the proposed method show a gradual decrease in packets retransmission rate as the number of nodes rose, DRCSN seems to have a packets retransmission rate less . in brief, DRCSN achieves between (68.33332%) and (17.894728%) packets retransmission rates when the number of nodes increases, while LELCR achieves between (82.49998%) and (29.16668%) packets retransmission rates when the number of nodes increases, while AODV-without the proposed method achieves between (92.5%) and (43.6111%) packets retransmission rates when the number of nodes increases.

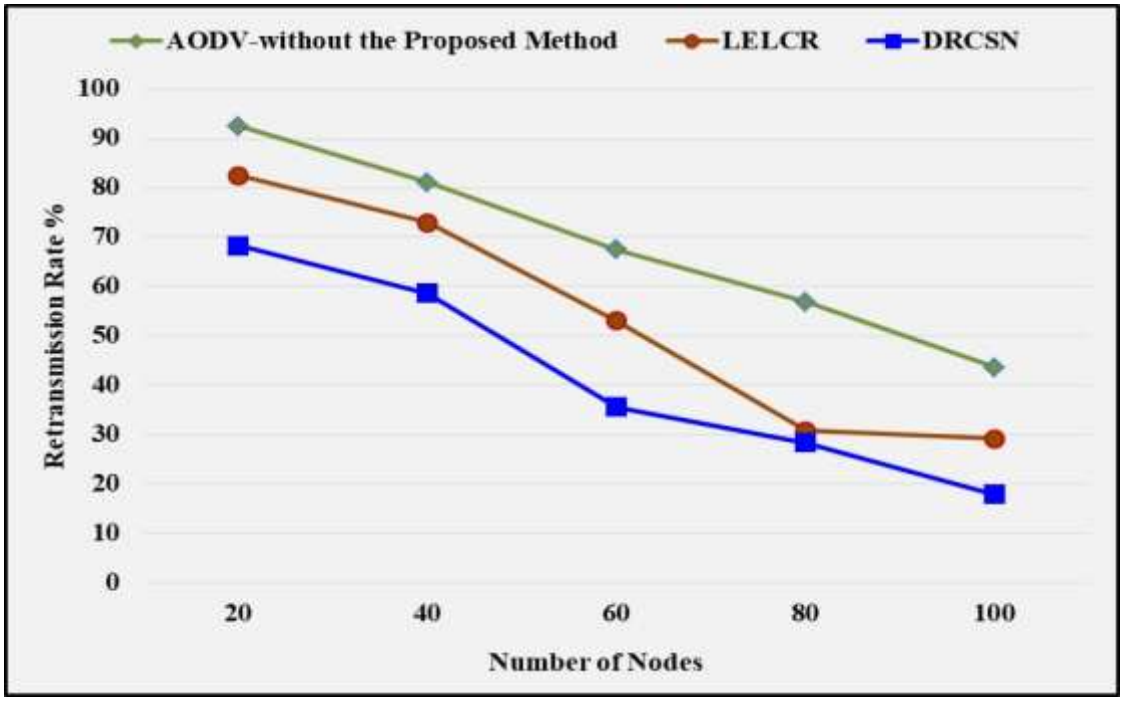

Figure 4.2 Impact of Number of Nodes vs PRR for DRCSN, LELCR and AODV-Without the Proposed Method

❖ Packet Delivery Ratio

Figure 4.3 is a depiction of a comparison of the performance of the DRCSN scheme, LELCR scheme and AODV-without the proposed method in terms of the packet delivery ratio, which is the percentage of total packets produced by the source node that has been received by the destination as compared to the total number of packets created by the source node. Although increasing the number of nodes will improve the packet delivery ratio for DRCSN, LELCR schemes, and AODV-without the proposed method, the performance of the DRCSN scheme has achieved better results.

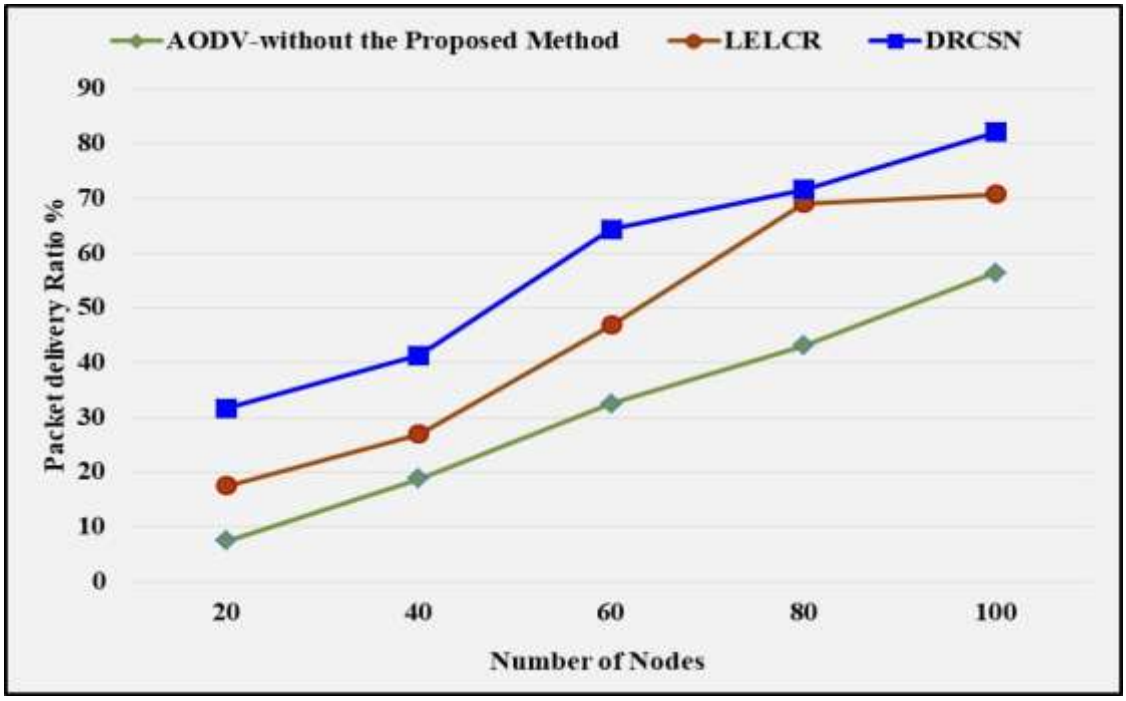

Figure 4.3 Impact of Number of Nodes vs Packet Delivery Ratio for DRCSN, LELCR and AODV-Without the Proposed Method

❖ Power Consumption

Figure 4.4 presents a comparison between the DRCSN scheme, LELCR scheme and AODV-without the proposed method in terms of their levels of performance with regard to power consumption. In the illustration that has been shown, the x-axis shows the number of nodes, while the y-axis represents the power consumption. Although the number of nodes rises, the

DRCSN scheme's power consumption value remains virtually the same as its initial value. This is the case even though the number of nodes increases. Because in the DRCSN, When selfish nodes exist, the intermediate nodes on the path to the destination will meet the requests without the need to return to the source and find an alternative path. So the number of next hops in the transmission will decrease, thus reduce energy consumption. While with the LELCR scheme, the values have increased from around 10% to 50%. While AODV-without the proposed method, the values have increased from around 13% to 63%.In terms of the amount of power consumption that is produced by the network, the performance of the DRCSN scheme is better.

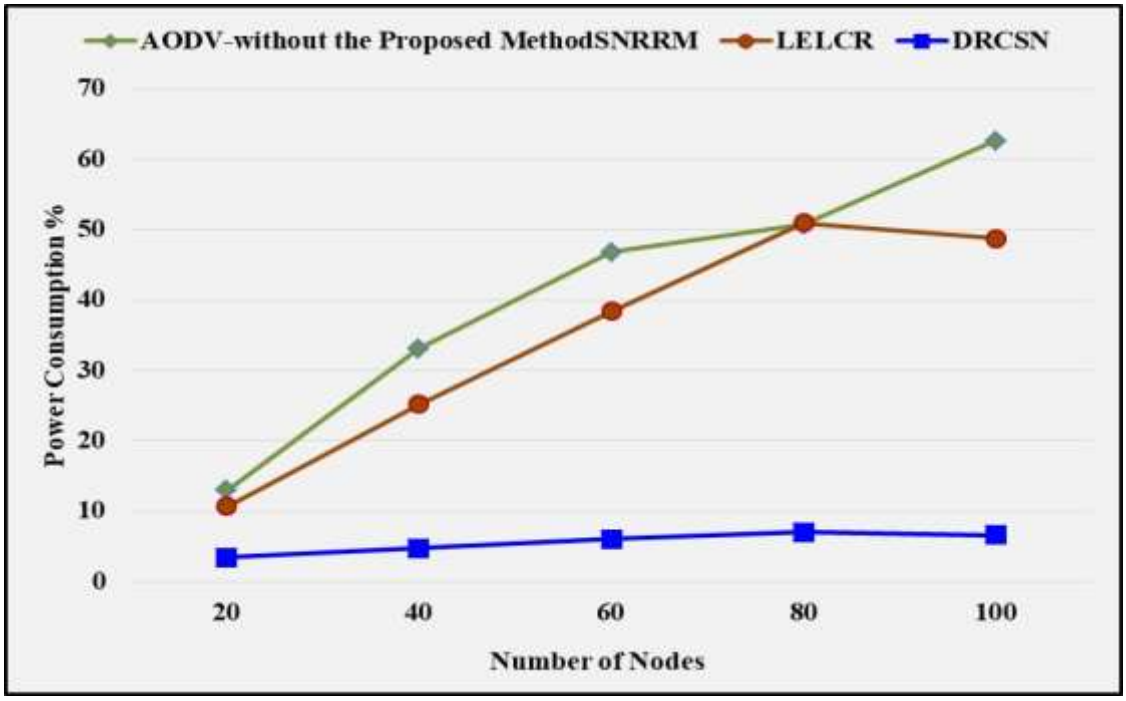

Figure 4.4 Impact of Number of Nodes vs Power Consumption for DRCSN, LELCR and AODV-Without the Proposed Method

❖ Average E2E Delay

The E2E delay can be described as the difference in time between the time moment at which the packet reaches the receiver and the time instant at which the packet is formed at the sender. The performance comparison of DRCSN scheme, LELCR scheme and AODV-without the proposed method in terms of

average E2E delay is shown in Figure 4.5. In the given figure, the x-axis shows the number of nodes, while the y-axis shows the average E2E delay. According to the obtained results, both DRCSN, LELCR schemes, and AODV-without the proposed method have linear is unstable. The range of values for the DRCSN may be found around (8 to 10s). However, The range of values for the LELCR is found to be around (43 to 55 s). While the range of values for the AODV-without the proposed method is found to be around (52 to 55 s). Because without using the proposed method, the source will continue to send packets to the destination, which continues to wait to receive the packets correctly. So, in terms of the average E2E delay, the performance of the DRCSN scheme is superior to that of the LELCR scheme and AODV-without the proposed method.

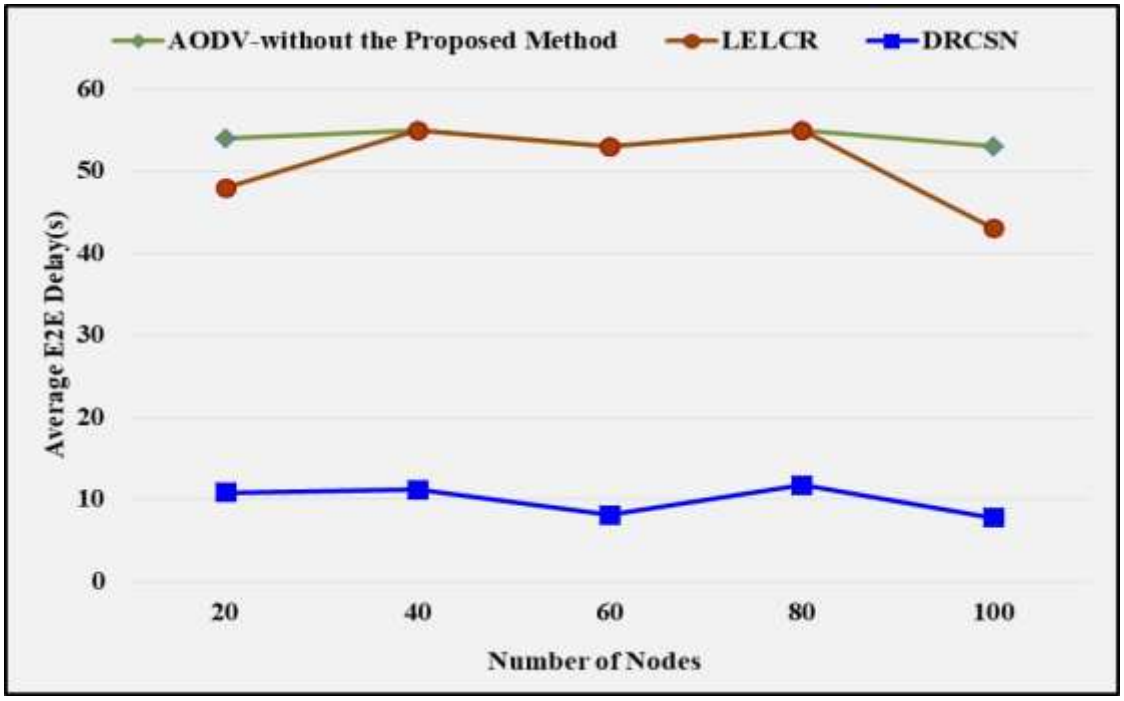

Figure 4.5 Impact of Number of Nodes vs Average E2E Delay for DRCSN, LELCR and AODV-Without the Proposed Method

In a conclusion, the LELCR scheme, DRCSN scheme and AODV-without the proposed method had been compared in different scenarios with an increase in the number of nodes from 20 to 100, and the results are presented

in Figures 4.1 to 4.5. Based on that, Table 4.2 illustrates the average value of throughput, power consumption, the packet delivery ratio, packets retransmission rate, and average E2E delay in all scenarios that tested both LELCR scheme, DRCSN scheme and AODV-without the proposed method in the case of the variety of the number of nodes. It seems that the DRCSN scheme has increased the throughput, and packet delivery ratio by 19% and 26%, respectively, compared to the LELCR scheme, and by 44%, and 84%, respectively, compared to AODV-without the proposed method. Also, it decreased the packets retransmission rate, average E2E delay, and power consumption by 22%, 80%, and 84%, respectively, compared to the LELCR scheme, and by 38%, 81% and 85%, respectively, from the AODV-without the proposed method as the DRCSN scheme's performance is much better than the LELCR scheme and AODV-without the proposed method.

Table 4.2 Average Results of Impact of a Number of Nodes for DRCSN, LELCR and AODV-Without the Proposed Method

Metrics	AODV-without the Proposed Method	LELCR	DRCSN
Throughput	١٧,٧٤٦٢٢٦٣kbps	**21.89736245**kbps	**25.97413348**kbps
Power Consumption	٤١,٢٧٩٦٣٦%	**34.827022%**	**5.5754468%**
Packet Delivery Ratio	٣١,٦٧٧٧٨%	**46.288896%**	**58.243288%**
Packets Retransmission Rate	**68.32222%**	**53.711544%**	**41.7567136%**
Average E2E Delay	٥٣,٩٨٨٤٣٦s	**50.788996s**	**9.9389784s**

4.5.2 Impact of a Variety of Speed of Nodes

To analyze the variety of speed nodes impact on the performance of DRCSN scheme, LELCR scheme and AODV-without the proposed method,

the maximum speed of nodes is varied as 5, 10, 15, 20, and 25 m/s, and the number of nodes is fixed at 100 nodes. Figures 4.6-4.10 shows the impact of a variety of speed nodes for DRCSN scheme, LELCR scheme and AODV-without the proposed method regarding to throughput, power consumption, packet delivery ratio, packets retransmission rate, and average E2E delay evaluation metrics.

- **Throughput**

Throughput is one of the important metrics of the network that determines the amount of the channel capacity that is really being utilized for productive transmission. Chooses a destination at the start of the simulation; this provides information on whether or not data packets were successfully delivered to their respective destinations. Figure 4.6 depicts the performance comparison of LELCR scheme, DRCSN scheme and AODV-without the proposed method in terms of throughput when the speed increases from 5 to 25.

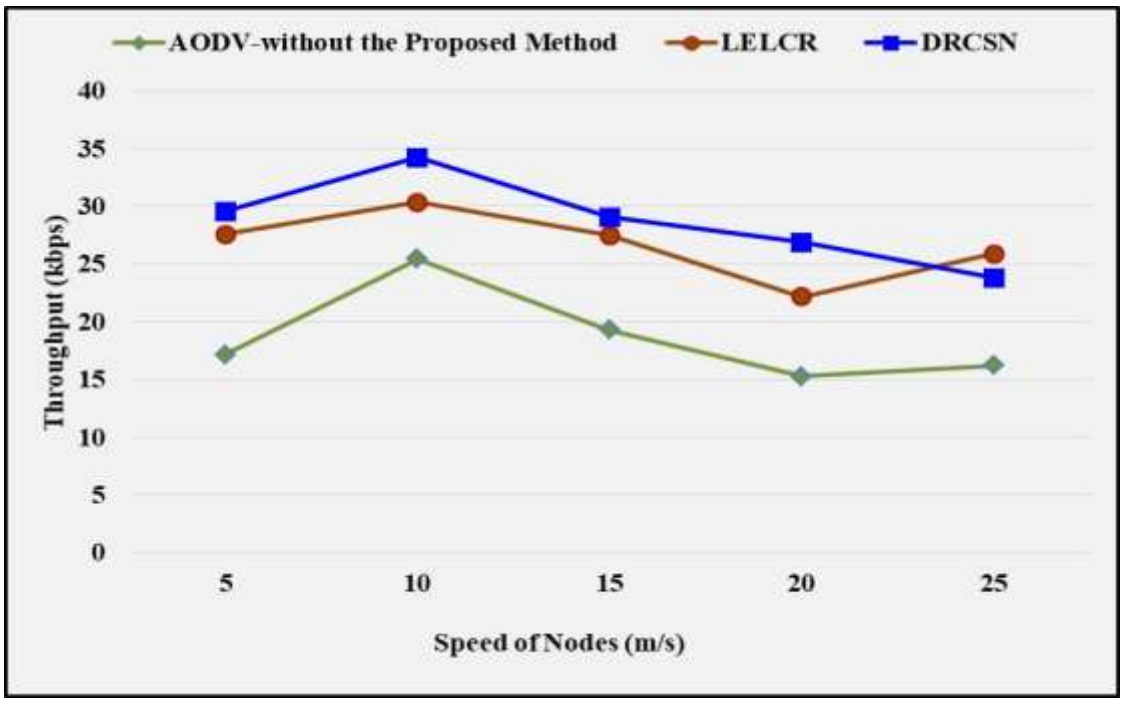

Figure 4.6 Impact of Variety of Speeds of Nodes vs Throughput for DRCSN, LELCR and AODV-Without the Proposed Method

In the given figure, when the speed is increased, the throughput of LELCR scheme, DRCSN scheme and AODV-without the proposed method will be decreased gradually. However, the DRCSN scheme is better than the LELCR scheme and AODV-without the proposed method in all cases..

❖ **Packet Delivery Ratio**

As mentioned previously, this metric refers to the proportion of the total number of data packets that have been sent from the source node to the destination node.

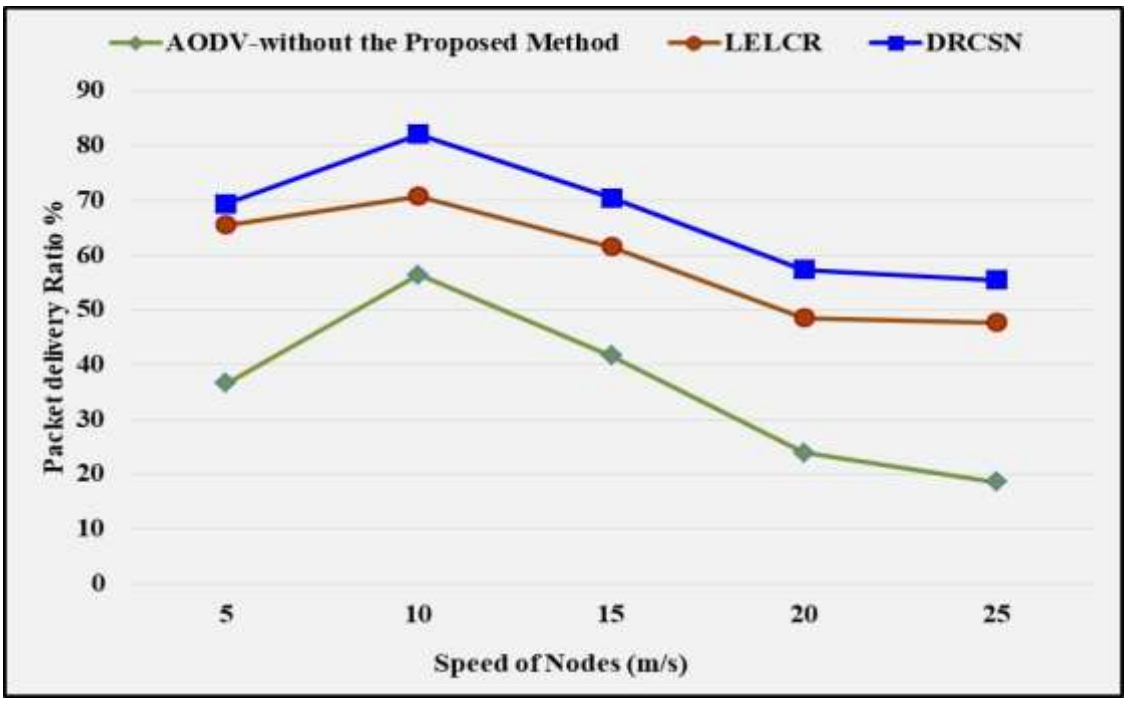

Figure 4.7 Impact of Variety of Speeds of Nodes vs Packet Delivery Ratio for DRCSN, LELCR and AODV-Without the Proposed Method

Figure 4.7 depicts the performance comparison of LELCR scheme, DRCSN scheme and AODV-without the proposed method in terms of packet delivery ratio over five different speeds. In the given figure, the x-axis represents the speed, whereas the y-axis represents the packet delivery ratio. When increasing the speed of nodes from 5 to 25 m/s, can be noticed that the packet delivery ratio of the DRCSN scheme is better than the LELCR scheme and AODV-without the proposed method. In brief, DRCSN achieves between

(82%) and (55%) packet delivery ratios when the speeds increase, while LELCR achieves between (71%) and (48%) when the speeds increase. however AODV-without the proposed method between (56%) and (18%) when the speeds increase.

❖ **Packets Retransmission Rate**

According to the obtained results, the packet retransmission rate of LELCR scheme, DRCSN scheme and AODV-without the proposed method over five different speeds is illustrated in Figure 4.8.

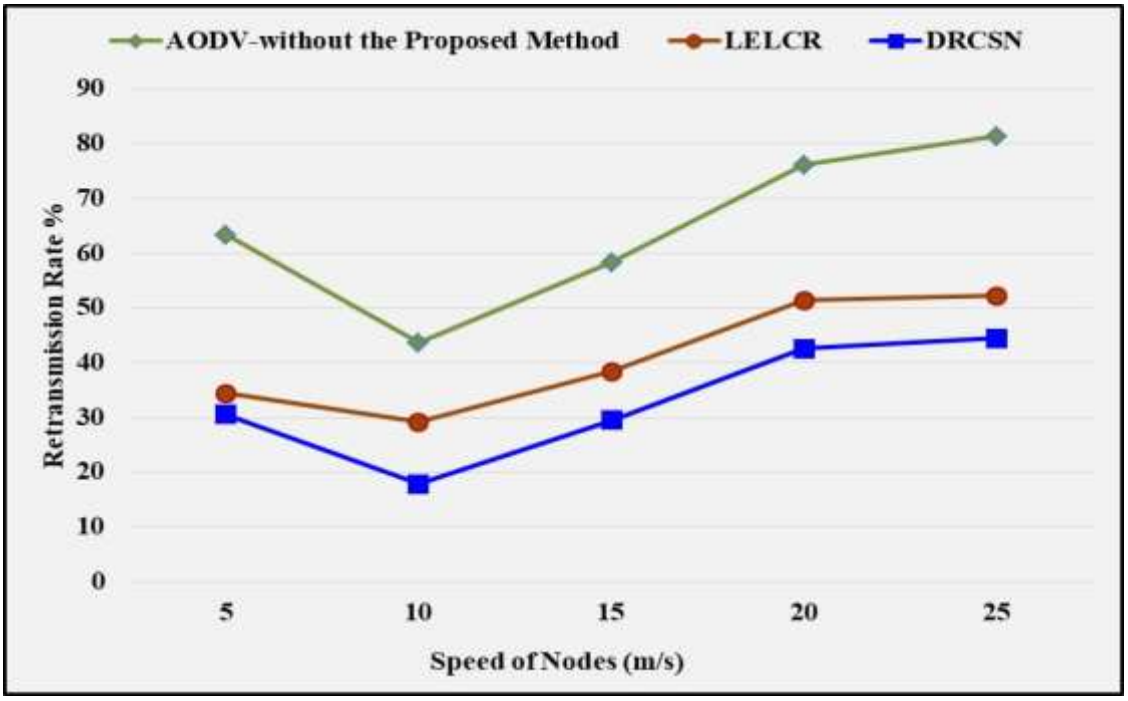

Figure 4.^ Impact of Variety of Speeds of Nodes vs PRR for DRCSN, LELCR and AODV-Without the Proposed Method

In the illustration, the x-axis shows the speed, while the y-axis represents the packets retransmission rate. Where the packets retransmission rate represents resending data packets that can not be sent successfully. So, the packets retransmission rate in DRCSN scheme is less than LELCR schem and AODV-without the proposed method. Therefore the performance of DRCSN scheme is better with an increased speed of nodes.

❖ Average E2E Delay

The performance comparison of LELCR scheme, DRCSN scheme and AODV-without the proposed method in terms of average E2E delay over five different speeds from 5 to 25 m/s is shown in Figure 4.9. According to the results obtained, can see that the LELCR scheme, AODV-without the proposed method and DRCSN scheme are completely different when it comes to average E2E delay. The range of values for the DRCSN may be found around (8s). However, the range of values for the LELCR is found to be around (43s). while the range of values for the AODV-without the proposed method is found to be around (53s).In the proposed method, the process of detecting selfish nodes early will avoid the destination waiting to receive data, This greatly reduces the average E2E delay. Therefore, this indicates that the performance of DRCSN is much better than ELCR and AODV-without the proposed method in terms of average E2E delay.

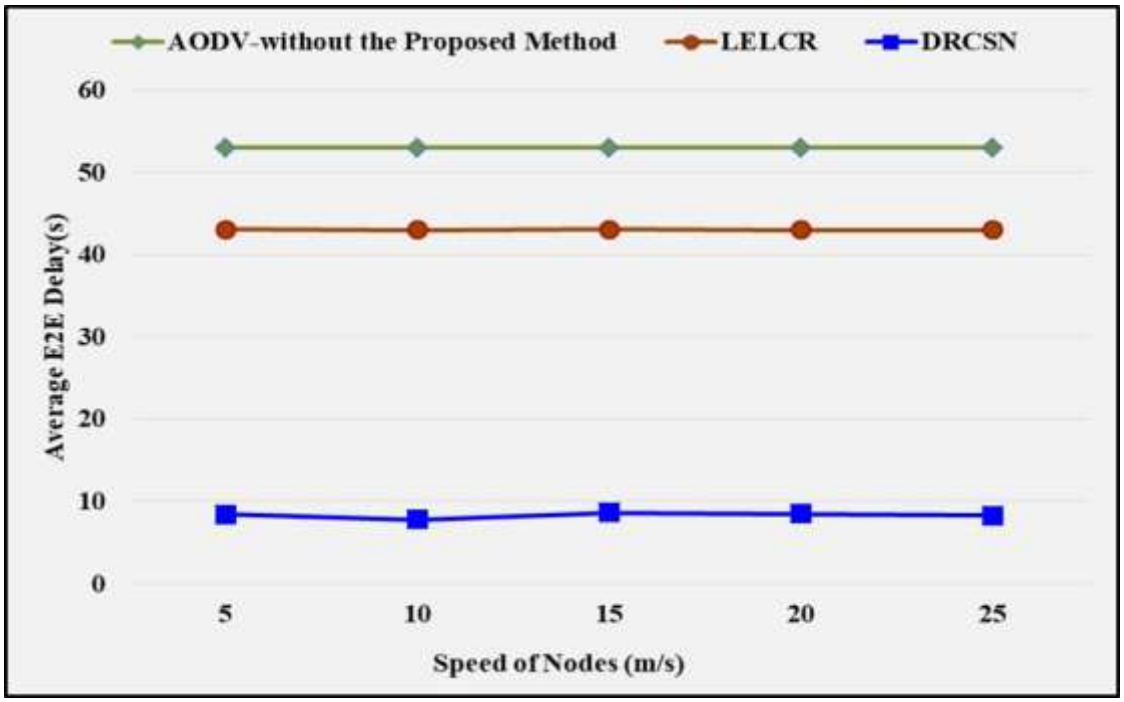

Figure 4.9 Impact of Variety of Speeds of Nodes vs Average E2E Delay for DRCSN, LELCR and AODV-Without the Proposed Method

❖ **Power Consumption**

As shown in Figure 4.10, the x-axis represents the speed, whereas the y-axis represents the power consumption. In the given figure, the performance comparison of DRCSN scheme, LELCR scheme and AODV-without the proposed method in terms of power consumption. Although all schemes have given linear results, it is noted that there are small changes in power consumption as speed increases. However, the performance of the DRCSN scheme is very better compared to the LELCR scheme and AODV-without the proposed method. Because to the fact that in the DRCSN, when selfish nodes are present, the intermediate nodes on the path to the destination will complete the requests without requiring a trip back to the source to choose a different path. As a result, there will be fewer next hops, which will result in less energy being used.

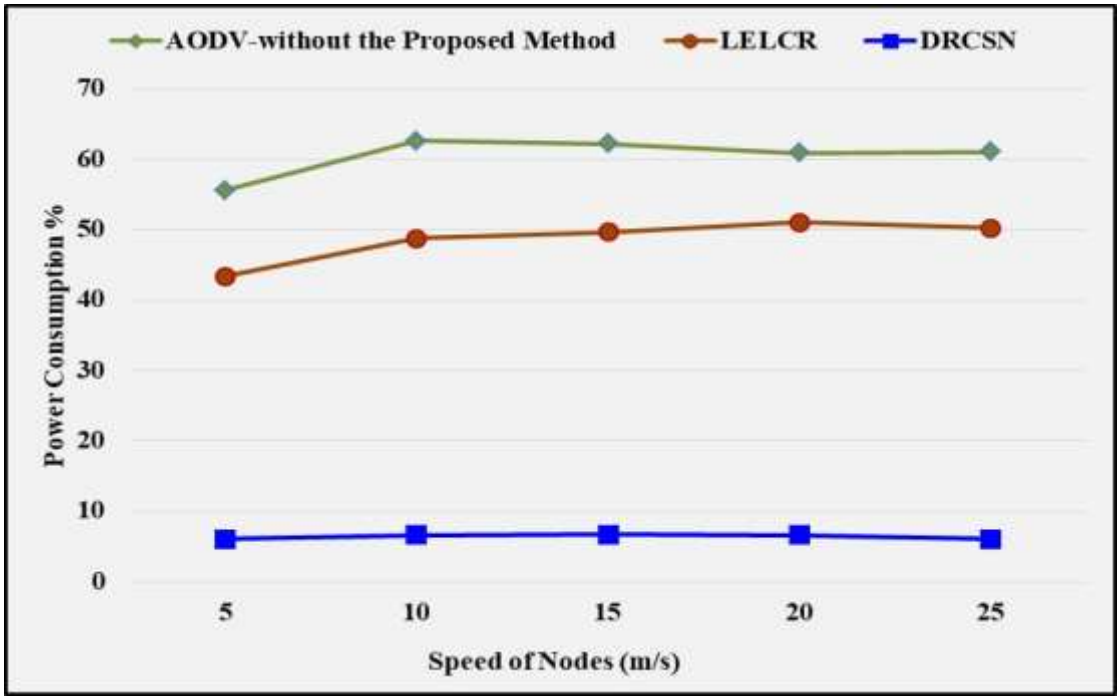

Figure 4.١٠ Impact of Variety of Speeds of Nodes vs Power Consumption for DRCSN, LELCR and AODV-Without the Proposed Method

In In a conclusion, the LELCR scheme, DRCSN scheme and AODV-without the proposed method had been compared in different scenarios with a

variety of speeds of nodes from 5 to 25 m/s, and the results are presented in Figures 4.6 to 4.10. Based on that, Table 4.3 displays the average value of throughput, the packet delivery ratio, packets retransmission rate, average E2E delay, and power consumption, in all scenarios that tested DRCSN, LELCR AODV-without the proposed method in the case of the variety of speed of nodes. It appears that the DRCSN scheme has increased the throughput and packet delivery ratio by 7% and 14%, respectively, compared to the LELCR scheme, and by 61%, and 91%, respectively, compared to AODV-without the proposed method. Furthermore, it decreased the packet retransmission rate, average E2E delay, and power consumption by 20%, 81%, and 87%, respectively, compared to the LELCR scheme., and by 65%, 85% and 88%, respectively, from the AODV-without the proposed method. Thus, the DRCSN scheme's performance is much better than the LELCR schemeand AODV-without the proposed method.

Table 4.3 Average Results of Impact of Variety of Speeds of Nodes for DRCSN, LELCR and AODV-Without the Proposed Method

Metrics	AODV-Without the Proposed Method	LELCR	DRCSN
Throughput	**18.68198918kbps**	**26.71381877kbps**	**28.71190121kbps**
Power Consumption	**60.533276%**	**48.647276%**	**6.4172%**
Packet Delivery Ratio	**35.4444412%**	**58.880704%**	**67%**
Packets Retransmission Rate	**64.55556%**	**41.119296%**	**33.0000016%**
Average End-to-End Delay	**52.994276s**	**43.034884s**	**8.2933184s**

4.6 Performance Evaluation of DRCSN, SNRRM and EBCS

In this section, the DRCSN scheme is implemented and compared with previous work involving SNRRM and EBCS using the NS-2 simulator. In order to compare the performance of DRCSN, SNRRM, and EBCS schemes, two scenarios have been considered (impact of number of nodes and impact of variety of speed of nodes).

4.6.1 Impact of a Number of Nodes

To analyze the number of nodes impact, the maximum number of nodes was varied as 20, 40, 60, 80, and 100 nodes and the speed node is fixed at 10m/s. Figures 4.11- 4.15 show the number of nodes impact on DRCSN, SNRRM, and EBCS schemes regarding the various performance metrics. As following:

- **Throughput**

Figure 4.11depicts the performance comparison of DRCSN and LELCR schemes in terms of throughput when the number of nodes increases from 20 to 100.

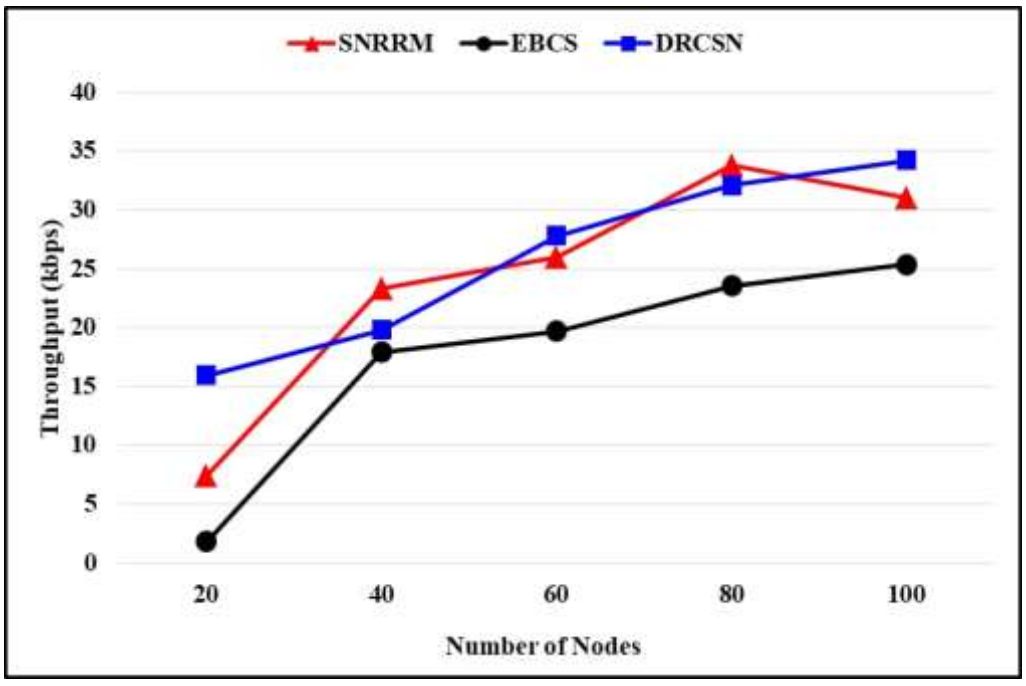

Figure 4.11 Impact of Number of Nodes vs Throughput for DRCSN, SNRRM, EBCS

As illustrated in the figure, when the number of nodes is increased, the throughput of DRCSN scheme will be superior to that of SNRRM and EBCS, respectively. Except in some cases, the throughput of SNRRM is better. Since the natural variations in the intermediate network and devices have an effect on the packets.

❖ Packets Retransmission Rate

In some cases, the sender nodes must resend packets that were lost or damaged during their initial transmission. This is one of the methods utilized to provide a reliable connection over a network.

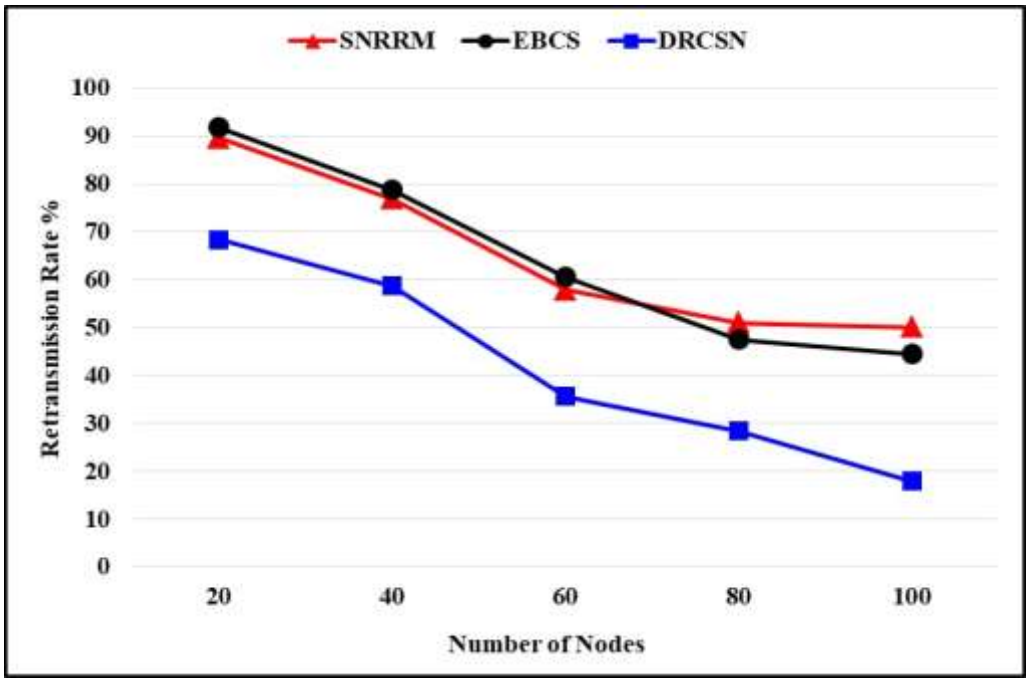

Figure 4.12 Impact of Number of Nodes vs PRR for DRCSN, SNRRM and EBCS

During simulations, came to the realization that the proportion of DRCSN retransmissions steadily decreased as the number of nodes increased as shown in Figure 4.12. When compared to previous related works, will find that DRCSN has the lowest packet retransmission rate since where the lower the retransmission of the packet rate, the better performance. can note that SNRRM has the highest retransmission rate and, therefore, the worst performance.

❖ Packet Delivery Ratio

As shown in Figure 4.13, the performance comparison of DRCSN, SNRRM, and EBCS schemes in terms of packet delivery ratio. In the illustration that has been shown, the x-axis shows the number of nodes, and the y-axis represents the packet delivery ratio. When the number of nodes increases, the recently introduced DRCSN scheme achieves a higher packet delivery ratio than the models that are used in the past SNRRM and EBCS. This indicates that the proposed scheme is a very efficient scheme.

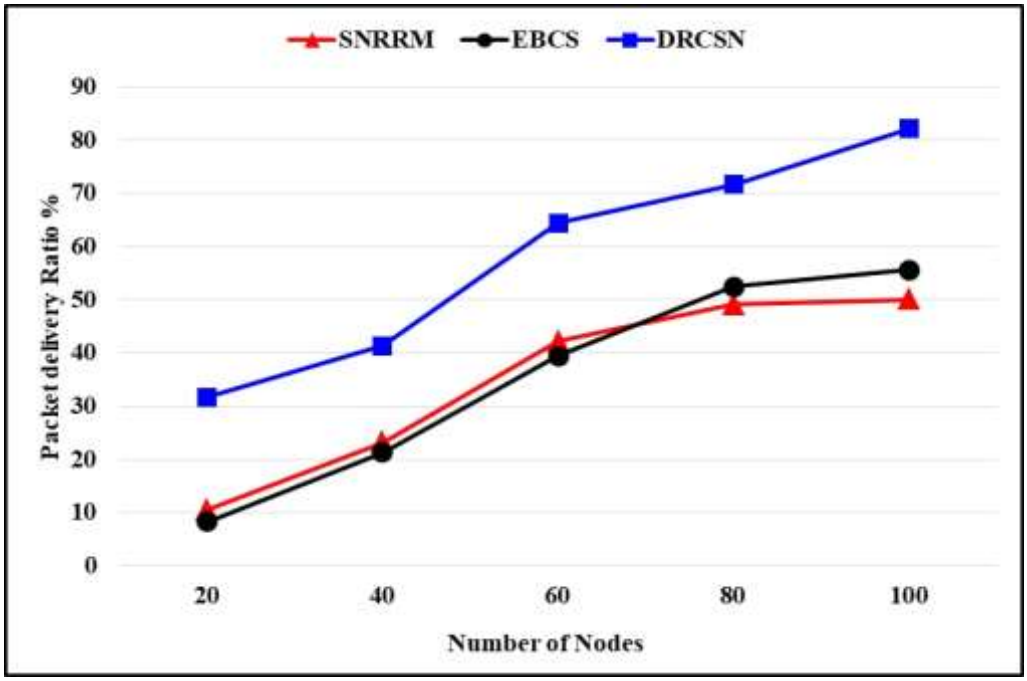

Figure 4.13 Impact of Number of Nodes vs Packet Delivery Ratio for DRCSN, SNRRM and EBCS

❖ Power Consumption

The performance comparison of DRCSN, SNRRM, and EBCS schemes in terms of power consumption is shown in Figure 4.14. In the given figure, the x-axis indicates the number of nodes, while the y-axis displays the amount of energy that has been residual. When the number of nodes rises, the value of the power consumption is almost never not more than 7% of its original value for DRCSN. When compared to other schemes, because in the SNRRM, and

EBCS the selfish nodes cause a decrease in the reception of packets to the destination, and therefore the source will resend the packets. So the number of next hops in the transmission will increase, which will lead to large consumption of energy. So can clearly notice that the proposed DRCSN scheme is the best, as it consumes the least amount of energy, while the SNRRM is the worst.

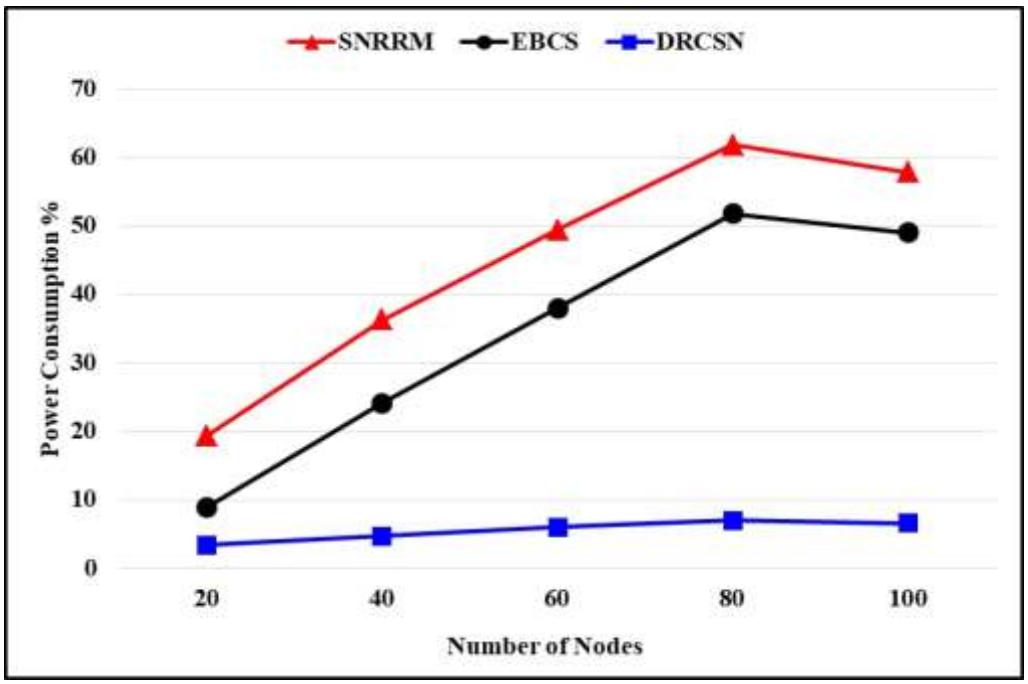

Figure 4.14 Impact of Number of Nodes vs Power Consumption for DRCSN, SNRRM and EBCS

❖ Average E2E delay

Figure 4.15 depicts the performance comparison of DRCSN, SNRRM, and EBCS schemes in terms of average E2E delay on a different number of nodes, 20 to 100 nodes. In the given figure, the x-axis represents the number of nodes, whereas the y-axis represents the average E2E delay. As note, the DECSN scheme has the lowest average E2E delay, which changes a little from 20 to 100 nodes. While in SNRRM, the average E2E delay is higher. This large difference in the average E2E delays, due to the previous methods

makes the source will keep sending packets to the destination, which will keep waiting for the packets to arrive correctly.

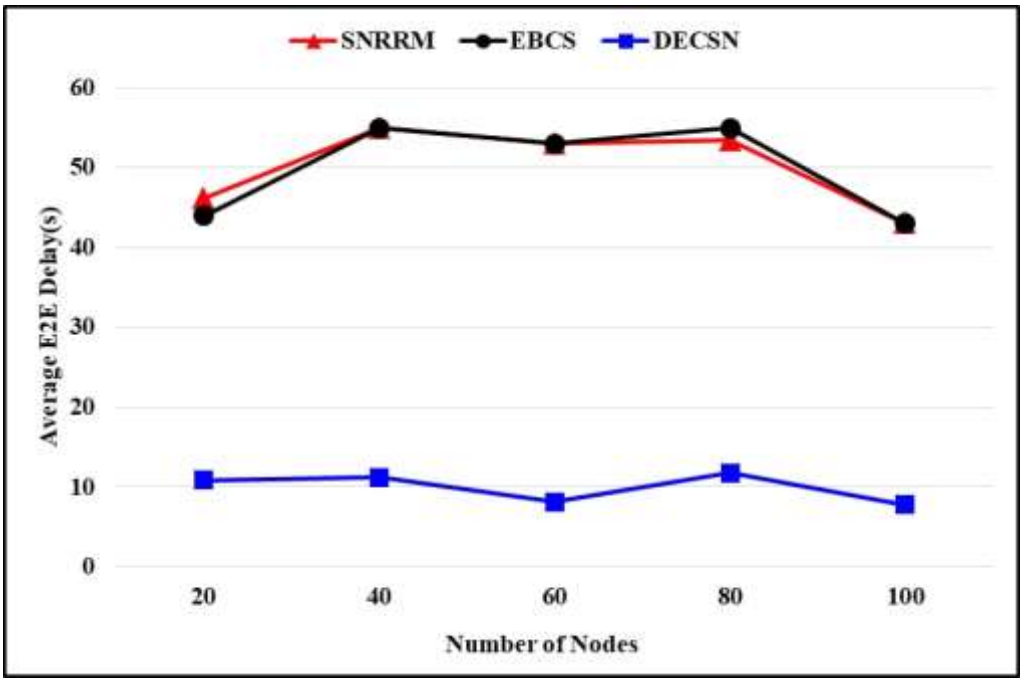

Figure 4.15 Impact of Number of Nodes vs Average E2E Delay for DRCSN, SNRRM and EBCS

As seen in the overall previous results (Figures 4.11-4.15), the proposed scheme DRCSN significantly outperforms the current schemes when increasing the number of nodes from 20 to 100 nodes. Table 4.4 displays the average results of throughput, the packet delivery ratio, packets retransmission rate, power consumption, and average E2E delay in all scenarios that tested DRCSN, SNRRM, and EBCS in the case of the number of nodes increasing from 20 to 100 nodes. According to scenarios, it appears that the DRCSN scheme has increased the throughput, and packet delivery ratio by 8%, and 66%, respectively, from the SNRRM scheme and by 44%, and 63%, respectively, from the EBCS scheme, also, It decreased the packets retransmission rate, average E2E delay, and power consumption by 35%, 80%, and 86%, respectively, from the SNRRM scheme, and by 35%, 79%, and 82%, respectively, from the EBCS scheme.

Table 4.4 Average Results of the Impact of Number of Nodes for DRCSN, SNRRM and EBCS

Metrics	SNRRM	EBCS	DRCSN
Throughput	24.29309765kbps	17.66971954kbps	25.97413348kbps
Power Consumption	44.973188%	34.4184952%	5.5754468%
Packet Delivery Ratio	34.9889%	35.4333332%	58.243288%
Packets Retransmission Rate	65.0111%	64.566668%	41.7567136%
Average End-to-End Delay	50.10858s	49.982976s	9.9389784s

4.6.2 Impact of a Variety of Speed Nodes

In this section, we will compare the results of the proposed scheme DRCSN with the SNRRM, and EBCS as well as analyze the impact of a variety of speed nodes on them. Where the maximum speed of nodes is varied as 5, 10, 15, 20, and 25 m/s and the number of nodes was fixed as 100 nodes. Figures 4.16-4.20 show the impact of a variety of speed of nodes for DRCSN, SNRRM, and EBCS regarding the various performance metrics.

- **Throughput**

Figure 4.16 depicts the performance comparison of DRCSN, SNRRM, and EBCS schemes in terms of throughput when the speed increases from 5 m/s to 25 m/s. In the given figure, the x-axis represents the number of speeds, whereas the y-axis represents the throughput values. When the speed is increased, the throughput of the DRCSN scheme will be better than SNRRM and EBCS, respectively. However, for some cases, the throughput of SNRRM will be better.

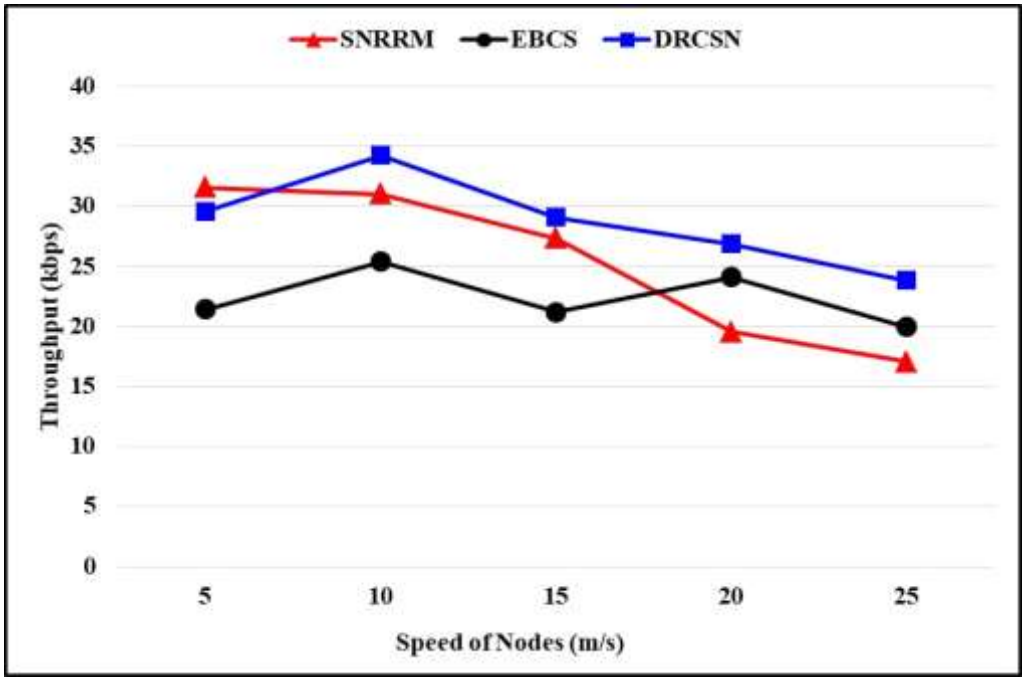

Figure 4.16 Impact of Variety of Speeds of Nodes vs Throughput for DRCSN, SNRRM, and EBCS

❖ Packet Delivery Ratio

As seen in Figure 4.17, the range of the packet delivery ratios varies based on the speed of nodes that is being used.

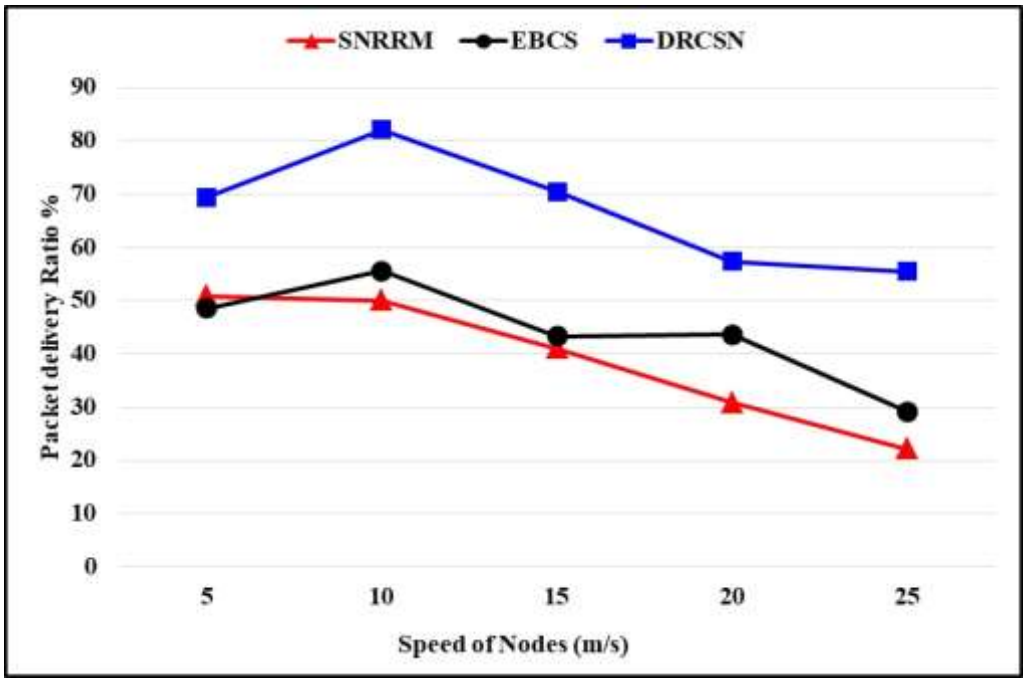

Figure 4.17 Impact of Variety of Speeds of Nodes vs Packet Delivery Ratio for DRCSN, SNRRM, and EBCS

This is something that may be seen; when its speed value is 10 m/s, the range is the greatest. This is in contrast to speeds of 20 m/s and 25 m/s, which have a range that gradually diminishes as they increase in speed. When comparing the DRCSN scheme of both SNRRM and EBCS,so find that the DRCSN scheme is the best in terms of delivery ratios, followed by EBCS and then SNRRM.

❖ Power Consumption

In terms of the power consumption in the network when the speed of nodes is increased from 5 m/s to 25 m/s, as be shown in Figure 4.18, where the x-axis represents the number of nodes, whereas the y-axis represents the residual energy.

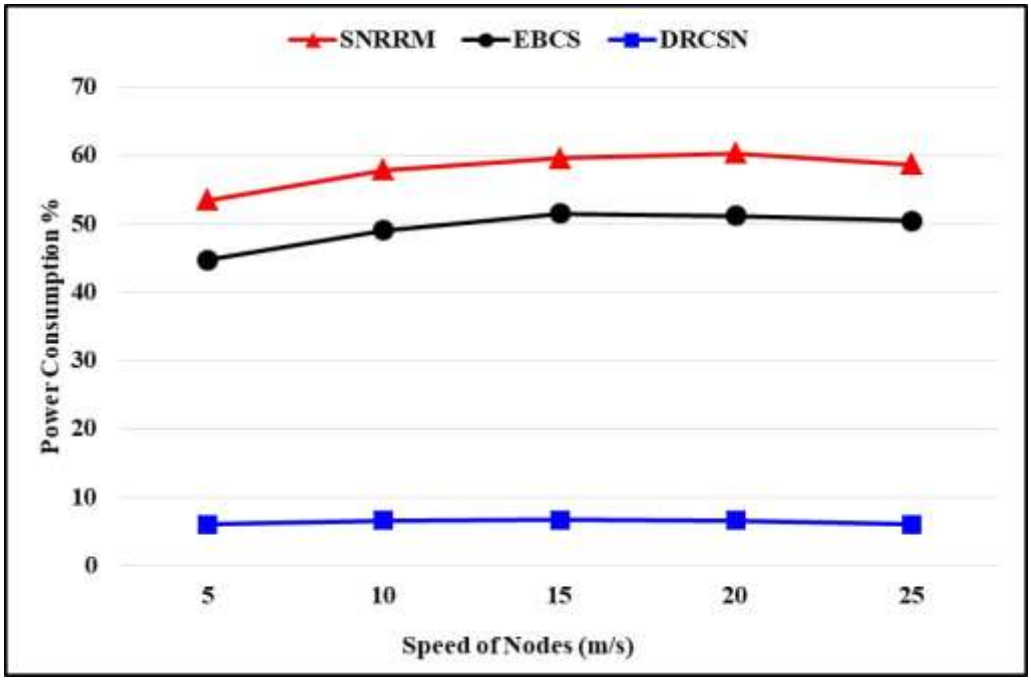

Figure 4.18 Impact of Variety of Speeds of Nodes vs Power Consumption for DRCSN, SNRRM, and EBCS

When performance comparison of DRCSN, SNRRM, and EBCS schemes in terms of power consumption. The performance of the DRCSN scheme is better compared to the SNRRM, and EBCS, respectively. Due to the fact that the intermediate nodes on the way to the destination in the DRCSN will finish

the requests without necessitating a journey back to the source to select an alternative path when selfish nodes are present. Because of this, there will be fewer next hops, which means less energy will be needed.

❖ Packets Retransmission Rate

As seen in Figure 4.19, the packet retransmission rate value for DRCSN ranged from around 30% to 44% when the speed values were modified from 5 m/s to 25 m/s. In contrast to the results obtained by previous methods, the DRCSN scheme produced much better results compared with the previous schemes SNRRM and EBCS, which ranged between (50% to 77%) and (44% to 70%), respectively.

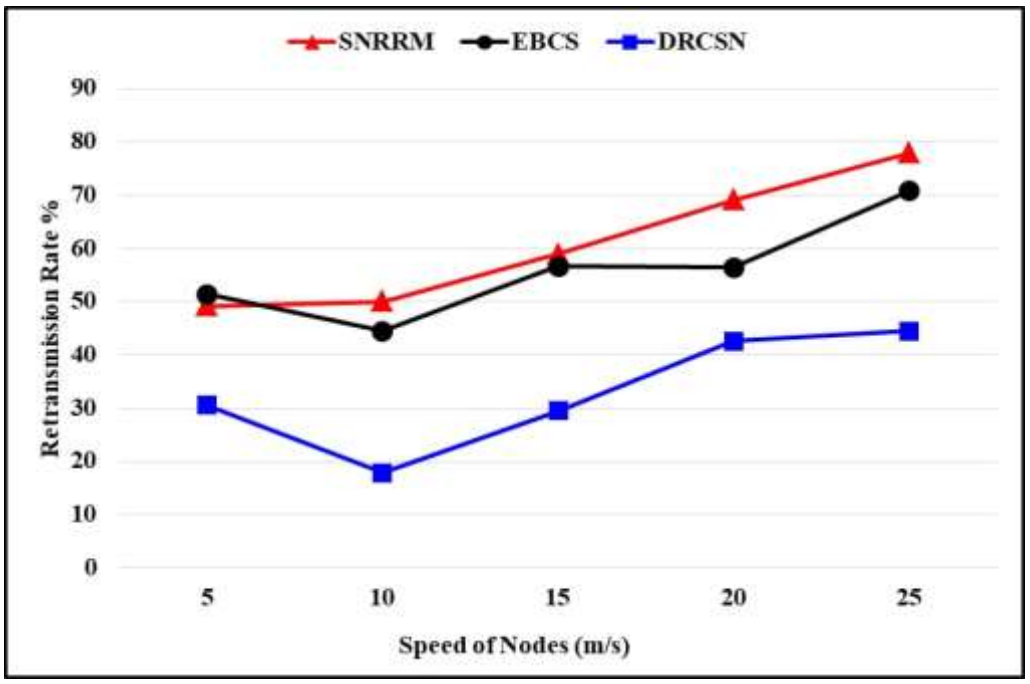

Figure 4.19 Impact of Variety of Speeds of Nodes vs PRR for DRCSN, SNRRM, and EBCS

❖ Average E2E Delay

The result is shown in Figure 4.20, the performance comparison of DRCSN, SNRRM, and EBCS schemes in terms of average E2E delay over five different speeds from 5 to 25 m/s. According to the obtained results, it is noted that the SNRRM and EBCS schemes are similar amounts when it

comes to average E2E delay for all settings of the speeds of nodes. On the other hand, the value of the average E2E delay for DRCSN is significantly different from theirs.That is because the source in SNRRM, and EBCS will continue sending packets to the destination, which will continue waiting for the packets to arrive correctly. So this will be indicated that the performance of DRCSN is much better than SNRRM, and EBCS.

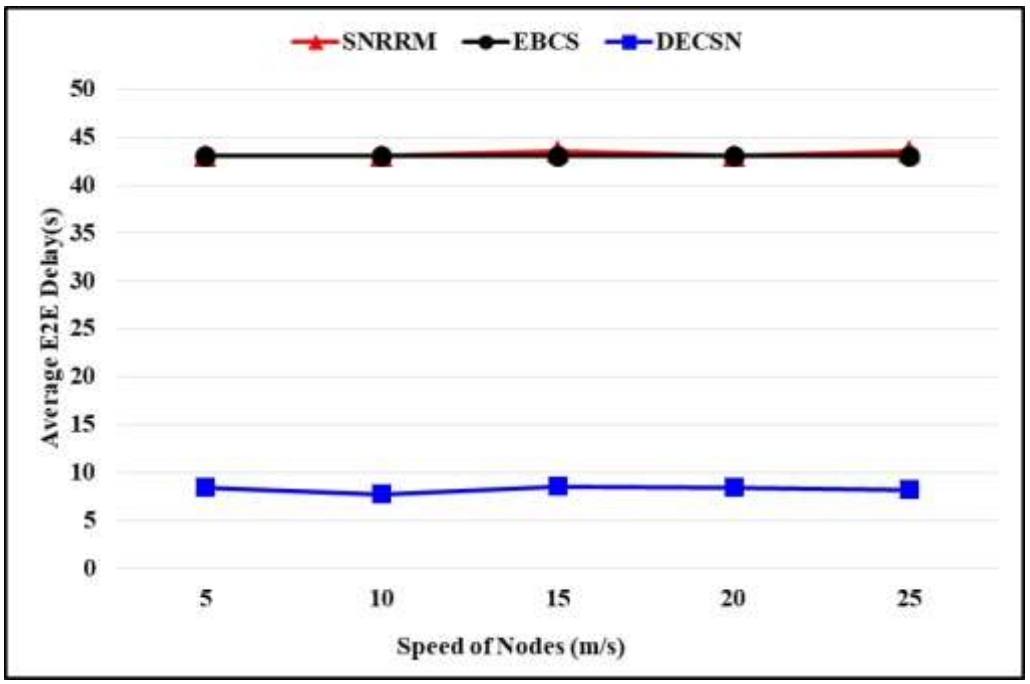

Figure 4.20 Impact of Variety of Speeds of Nodes vs Average E2E Delay for DRCSN, SNRRM, and EBCS

In a conclusion, the DRCSN, SNRRM, and EBCS schemes had been compared in different scenarios with a variety of speeds of nodes from 5 to 25 m/s. It appears that the proposed scheme, DRCSN was significantly outperformed in all tests. Table 4.5 displays the average results of throughput, the packet delivery ratio, packets retransmission rate, power consumption, and average E2E delay in all scenarios that tested DRCSN, SNRRM, and EBCS in the case of the number of nodes increasing from 20 to 100 nodes. According to scenarios, the DRCSN scheme has increased the throughput, and packet delivery ratio by 16% and 71%, respectively, from the SNRRM

scheme and by 31% and 52%, respectively, from the EBCS scheme, also decreases the packets retransmission rate and average E2E delay and power consumption by 46%, 81 %, and 89% respectively from the SNRRM scheme, and by 45%, 80%, and 86% respectively, from the EBCS scheme.

Table 4.5 Average Results of Impact of Variety for Speeds of Nodes of DRCSN, SNRRM and EBCS

Metrics	SNRRM	EBCS	DRCSN
Throughput	**25.29581782kbps**	**22.40940549kbps**	**28.71190121kbps**
Power Consumption	**57.95892%**	**49.368084%**	**6.4172%**
Packet Delivery Ratio	**38.9649252%**	**44.05556%**	**67%**
Packets Retransmission Rate	**61.035076%**	**55.94444%**	**33.0000016%**
Average End to End Delay	**43.194072s**	**42.994304s**	**8.2933184s**

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusion

The optimization method suggested in this thesis improves the selfish nodes in the MANET and makes them cooperative to the maximum extent. In addition, the AODV protocol as the MANET protocol had been developed by employing the proposed method inside it. NS-2 was used for the simulation of the proposed method. Many scenarios have been created by changing the network parameters (number of nodes, speed of nodes) sequentially. AWK is used to read trace files for each scenario and create a file for each scenario that contains network evaluation metrics (throughput, power consumption, packet delivery ratio, packets retransmission rate, average E2E delay). The following conclusions can be drawn as a result of the implementation:

- The obtained results demonstrate that the proposed method obtained the best results for the two scenarios (the number of nodes, and speed of nodes). When compared with current methods (standard AODV, SNRRM, and EBCS), the proposed method (DRCSN) significantly outperformed when increasing the number of nodes from 20-100 nodes. DRCSN had increased the throughput by 8%, 44%, and 44% from the SNRRM, EBCS, and the standard AODV, respectively. Furthermore, it had improved the packet delivery ratio by 66%, 63%, and 87% from the SNRRM, EBCS, and the standard AODV, respectively. On the other hand, DRCSN decreased the packet retransmission rate by 35%, 35%, and 38% from the SNRRM, EBCS, and the standard AODV, respectively. While in the case of average E2E delay, it decreased by 80%,79%, and 81% from the SNRRM, EBCS, and the standard AODV, respectively. It is observed also, the proposed method decreased the power consumption by 86%,82%, and 85% compared with the SNRRM, EBCS, and the standard AODV, respectively.

- DRCSN method significantly outperformed the SNRRM and EBCS with a variety of speed of nodes from 5-25 m/s. It increased the throughput by 16%, 31%, and 61% from the SNRRM, EBCS, and the standard AODV, respectively. Also, it had increased the packet delivery ratio by 71%, 52%, and 91% from the SNRRM, EBCS, and the standard AODV, respectively. On the other hand, it decreased the packet retransmission rate by 46%,45%, and 65% from the SNRRM, EBCS, and the standard AODV, respectively. While in the case of average E2E delay, DRCSN decreased by 81%,80%, and 85% from the SNRRM, EBCS, and the standard AODV, respectively. Finally, the proposed method decreased power consumption by 89%,86%, and 88% from the SNRRM, EBCS, and the standard AODV, respectively. As a final result, the proposed method achieved its purpose, which is to detect, and reintroduce the selfish node to the network and force it to collaborate. Hence, it enhanced the performance of MANET.

5.2 Limitation

In this section, some of the limitations are presented which had not been considered in the proposed method.

1. Security is not taken into consideration in the proposed method as malicious nodes, Blackhole attack. Security is a paramount concern in MANET because of its intrinsic vulnerabilities.
2. Other important MANET parameters that may affect the behaviour of the network will not also be considered, such as the number of connections, the difference in terrain area.
3. The scalability issue.

5.3 Future work

Our proposed method can be extended as future work in four points:

1. Taking malicious nodes into consideration; black hole and gray hole attacks in MANET.
2. It takes into consideration MANET parameters that may affect network behaviour, such as the number of connections and different terrain areas.
3. Apply proposed method with other MANET protocols, such as the DSR or DSDV.
4. Implement the proposed method over large-scale networks.

REFERENCES

[1] R. Kaur, R. Singla, B. Kaur, and S. Singh, "MANETs: Overview, Tools, Security and Applications in Health Care," *Aust. J. BASIC Appl. Sci.*, vol. 11, no. 8, pp. 1–6, 2017, [Online]. Available: http://www.ajbasweb.com/old/ajbas/2017/Special issue (iSTEM '17)/1-6.pdf.

[2] K. Susan, K. C., J.-O. A. M., and M. E. S., "An Improved Token-Based Umpiring Technique for Detecting and Eliminating Selfish Nodes in Mobile Ad-hoc Networks," *Egypt. Comput. Sci. J.*, vol. 44, no. 2, pp. 74--85, 2020, [Online]. Available: shttps://www.researchgate.net/publication/341597901%0A.

[3] K. Saharan and H. Pande, "A Survey on Energy Efficient Roution Protocols for MANET," *Int. J. Adv. Eng. Technol.*, vol. 6, no. 1, pp. 370–380, 2013, [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.384.7699&rep=rep1&type=pdf.

[4] R.-I. Ciobanu, C. Dobre, M. Dascalu, S. Trausan-Matu, and V. Cristea, "Collaborative Selfish Node Detection with an Incentive Mechanism for Opportunistic Networks," *IFIP/IEEE IM2013 Work. 5th Int. Work. Manag. ofthe Futur. Internet*, pp. 1161–1166, 2014, doi: 10.1016/j.jnca.2014.01.009.

[5] S. Kumar, K. Dutta, and G. Sharma, "A Detailed Survey on Selfish Node Detection Techniques for Mobile Ad Hoc Networks," *2016 4th Int. Conf. Parallel, Distrib. Grid Comput. PDGC 2016*, vol. 3, pp. 122–127, 2016, doi: 10.1109/PDGC.2016.7913128.

[6] H. Yadav and H. K. Pati, "A Survey on Selfish Node Detection in MANET," *Proc. - IEEE 2018 Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2018*, pp. 217–221, 2018, doi:

10.1109/ICACCCN.2018.8748420.

[7] S.Senthilkumar and J.William, “A Survey on Reputation Based Selfish Node Detection Techniques in Mobile Ad Hoc Network,” *J. Theor. Appl. Inf. Technol.*, vol. 60, no. 2, pp. 208–215, 2014, [Online]. Available: https://www.researchgate.net/publication/289328212_A_survey_on_reputation_based_selfish_node_detection_techniques_in_mobile_Ad_Hoc_network.

[8] J. M. S. P. Josh Kumar, A. Kathirvel, N. Kirubakaran, P. Sivaraman, and M. Subramaniam, “A Unified Approach for Detecting and Eliminating Selfish Nodes in MANETs Using TBUT,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2015, no. 1, pp. 1–11, 2015, doi: 10.1186/s13638-015-0370-x.

[9] S. Janakiraman and M. Rajendiran, “An Erlang Factor-Based Conditional Reliability Mechanism for Enforcing Co-operation in MANETs,” *Serbian J. Electr. Eng.*, vol. 13, no. 2, pp. 265–284, 2016, doi: 10.2298/SJEE1602265J.

[10] J. Sengathir and R. Manoharan, “Exponential Reliability Factor Based Mitigation Mechanism for Selfish Nodes in MANETs,” *J. Engg*, vol. 4, no. 1, pp. 44–64, 2016.

[11] P. Prasath and P. . Scholar, “Detecting Selfish nodes in MANET Using Record- Trust Based- Detection with Collaborative Watchdog,” *Int. J. Eng. Res. Technol.*, vol. 4, no. 11, pp. 1–5, 2016.

[12] S. B. B, I. K, and A. Kumar, “Detection of Selfish and Malicious Node in Mobile Ad-Hoc Network with NS-2 Using Chord Algorithm,” *Int. J. Eng. Technol.*, vol. 9, no. 2, pp. 466–471, 2017, doi: 10.21817/ijet/2017/v9i2/170902326.

[13] S. Sayyar, A. Khan, F. Ullah, H. Anwar, and Z. Kaleem, “Enhanced TWOACK Based AODV Protocol for Intrusion Detection System,” in

2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development, iCoMET 2018 - Proceedings, 2018, pp. 1–4, doi: 10.1109/ICOMET.2018.8346444.

[14] S. J. Patil, "Secure Collaborative Contact Based Watchdog for Detecting Selfish Nodes in MANET Using Hash Chain Technique," *Int. J. Adv. Res. Comput. Commun. Eng. IJARCCE*, vol. 7, no. 6, pp. 179–184, 2018, doi: 10.17148/IJARCCE.2018.7630.

[15] G. Narayanan, J. K. Das, M. Rajeswari, and R. S. Kumar, "Game Theoretical Approach with Audit Based Misbehavior Detection System," in *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, 2018, pp. 1932–1935, doi: 10.1109/ICICCT.2018.8473197.

[16] S. Mubeen and S. Johar, "Detection and Elimination of the Selfish Node in Ad-Hoc Network Using Energy Credit Based System," *J. Netw. Inf. Secur.*, vol. 7, no. 2, pp. 18–22, 2019, [Online]. Available: http://www.publishingindia.com/jnis.

[17] M. Mohamed Musthafa, K. Vanitha, A. M. J. M. D. Zubair Rahman, and K. Anitha, "An Efficient Approach to Identify Selfish Node in MANET," in *2020 International Conference on Computer Communication and Informatics, ICCCI 2020*, 2020, pp. 1–3, doi: 10.1109/ICCCI48352.2020.9104076.

[18] R. Mangayarkarasi and R. Manikandan, "Cost Effective Collaborative Anomaly Detection System for Selfish Node Attacks in MANET," *J. Crit. Rev.*, vol. 7, no. 13, pp. 148–154, 2020, doi: 10.31838/jcr.07.13.26.

[19] M. Ponnusamya, A. Senthilkumarb, and R.Manikandan, "Detection of Selfish Nodes Through Reputation Model In Mobile Adhoc Network - MANET," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 9, pp. 2404–2410–2404– 2410, 2021, [Online]. Available:

https://www.turcomat.org/index.php/turkbilmat/article/view/3720.

[20] B. H. Alqarni and A. S. Almogren, “Reliable and Energy Efficient Protocol for MANET Multicasting,” *J. Comput. Networks Commun.*, vol. 2016, pp. 1–13, 2016, doi: 10.1155/2016/9146168.

[21] M. Mohamed Musthafa, K. Vanitha, A. M. J. M. D. Zubair Rahman, and K. Anitha, “An Efficient Approach to Identify Selfish Node in MANET,” in *2020 International Conference on Computer Communication and Informatics, ICCCI 2020*, 2020, pp. 1–3, doi: 10.1109/ICCCI48352.2020.9104076.

[22] M. A. Al-Absi, A. A. Al-Absi, M. Sain, and H. Lee, “Moving Ad Hoc Networks—A Comparative Study Mohammed,” *Multidiscip. Digit. Publ. Inst. MDPI*, vol. 13, no. 11, pp. 1–31, 2021, doi: 10.3390/su13116187.

[23] A. R. Sobhy, M. M. Elfaham, and A. Hashad, “Fanet Cloud Computing,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 10, pp. 88–93, 2016.

[24] P. Mohapatra and S. V.krisgnamurthy, “AD Hoc Networks Technologies and Protocols,” United States of America: Springer US, 2005, p. 268.

[25] I. Stojmenovic, *Handbook of Wireless Networks and Mobile Computing*. Wiley-Interscience, 2002.

[26] Y. Wang and F. Li, “Guide to Wireless Ad Hoc Networks,” *Comput. Commun. Networks*, pp. 503–525, 2009, doi: 10.1007/978-1-84800-328-6.

[27] S. Yousefi, M. S. Mousavi, and M. Fathy, “Vehicular Ad hoc Networks (VANETs): Challenges and Perspectives,” *ITST 2006 - 2006 6th Int. Conf. ITS Telecommun. Proc.*, pp. 761–766, 2006, doi: 10.1109/ITST.2006.289012.

[28] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying Ad-Hoc Networks

(FANETs): a Survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013, doi: 10.1016/j.adhoc.2012.12.004.

[29] A. Bakshi, A. K. Sharma, and A. Mishra, "Significance of Mobile AD-HOC Networks (MANETS)," *Int. J. Innov. Technol. Explor. Eng.*, vol. 4, no. 2, pp. 2278–3075, 2013, [Online]. Available: https://pdfs.semanticscholar.org/bbb2/2fbaecce3bb6be0814c0d87eee89f867cd2c.pdf.

[30] J. Loo, J. L. Mauri, and J. H. Ortiz, *Mobile Ad Hoc Networks Current Status and Future Trends*. 2004.

[31] S. K. Sakalabattula and S. S. Kumar, "Overview (Advantages and Routing Protocols) of MANET," *Adv. Comput. Sci. Technol.*, vol. 10, no. 5, pp. 855–861, 2017, [Online]. Available: https://www.ripublication.com/acst17/acstv10n5_19.pdf.

[32] K. Muralidhar and K. Madhavi, "An Investigation into the Operational Limitations of Mobile Ad Hoc Networks," *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, pp. 1373–1376, 2018, doi: 10.1109/WiSPNET.2017.8299988.

[33] R. Alubady, M. Al-Samman, A. Habbal, S. Hassan, and S. Arif, "Performance Analysis of Reactive and Proactive Routing Protocols in MANET," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 3, pp. 1468–1478, 2015.

[34] N. Swami and A. Bairwa, "A Literature Survey of MANET," *Int. Res. J. Eng. Technol.*, vol. 03, no. 02, pp. 11–14, 2016.

[35] C. E. Perkins and P. Bhagwat, "Highly Dynamic (DSDV) for Mobile Computers Routing," *Proc. ACM SIGCOMM94, London, UK*, vol. 24, no. 4, pp. 234–244, 1994, doi: https://doi.org/10.1145/190314.190336.

[36] T. Clausen *et al.*, "Optimized Link State Routing Protocol (OLSR)," *Netw. Work. Group. 2003*, pp. 1–53, 2003, [Online]. Available: https://www.researchgate.net/publication/277255608_Optimized_link_s

tate_routing_protocol_OLSR.

[37] S. Murthy and J. J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Networks,” *Mob. Networks Appl.*, vol. 1, no. 2, pp. 183–197, 1996, doi: 10.1007/BF01193336.

[38] N. Garg, K. Aswal, and D. C. Dobhal, “A Review of Routing Protocols in Mobile Ad Hoc Networks,” *Int. J. Inf. Technol. Knowl. Manag.*, vol. 5, no. 1, pp. 177–180, 2012.

[39] S. Deepak and H. Anandakumar, “AODV Route Discovery and Route Maintenance in MANETs,” *2019 5th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2019*, pp. 1187–1191, 2019, doi: 10.1109/ICACCS.2019.8728456.

[40] D. B. Johnson and D. A. Maltz, “DSR : The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks,” *Comput. Sci. Dep. Carnegie Mellon Univ. Addison-Wesley*, no. January 2002, pp. 139–172, 1996, [Online]. Available: http://www.monarch.cs.cmu.edu/.

[41] K. H. Lim, “Performance Enhancement of the Temporally-Ordered Routing Algorithm,” 2007.

[42] D. E. Mustafa Ahmed and O. O. Khalifa, “A Comprehensive Classification of MANETs Routing Protocols,” *Int. J. Comput. Appl. Technol. Res.*, vol. 6, no. 3, pp. 141–158, 2017, doi: 10.7753/ijcatr0603.1004.

[43] N. Beijar, “Zone Routing Protocol (ZRP),” *Netw. Lab. Helsinki Univ. Technol. Finl.*, pp. 1–12, 2002.

[44] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, “SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks,” *Proc. Int. Symp. Mob. Ad Hoc Netw. Comput.*, pp. 303–314, 2003.

[45] M. V. Narayana and A. Atmakuri, “A-ZHLS: Adaptive ZHLS Routing Protocol for Heterogeneous Mobile Adhoc Networks,” *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 1626–1630, 2018, doi:

10.14419/ijet.v7i3.14242.

[46] A. Bagwari, R. Jee, P. Joshi, and S. Bisht, “Performance of AODV Routing Protocol With Increasing the MANET Nodes and its Effects on QoS of Mobile Ad Hoc Networks,” *Proc. - Int. Conf. Commun. Syst. Netw. Technol. CSNT 2012*, pp. 320–324, 2012, doi: 10.1109/CSNT.2012.76.

[47] M. Manjunath and D. H. Manjaiah, “Comparative Study of AODV, SAODV, DSDV and AOMDV Routing Protocols in MANET Using NS2,” *Int. J. Commun. Netw. Syst.*, vol. 004, no. 001, pp. 18–22, 2015, doi: 10.20894/ijcnes.103.004.001.005.

[48] A. M. El-Semary and H. Diab, “BP-AODV: Blackhole Protected AODV Routing Protocol for MANETs Based on Chaotic Map,” *IEEE Access*, vol. 7, pp. 95197–95211, 2019, doi: 10.1109/ACCESS.2019.2928804.

[49] K. E.B., “Ad hoc On-Demand Distance Vector (AODV) Routing Status,” *Kaos GL Derg.*, no. 76, pp. 147–173, 2003.

[50] H. Mustafa and N. A. Noureldien, “Detection of Route Discovery Misbehaving Nodes in AODV MANETs: A Survey,” *Int. J. Networks Commun.*, vol. 8, no. 4, pp. 115–122, 2018, doi: 10.5923/j.ijnc.20180804.03.

[51] J. Sengathir and R. Manoharan, “Exponential Reliability Factor Based Mitigation Mechanism for Selfish Nodes in MANETs,” *J. Engg*, vol. 4, no. 1, pp. 44–64, 2016.

[52] S. Aifa and T. Thomas, “Review on Different Techniques Used in Selfish Node Detection,” in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology, ICCSDET 2018*, 2018, pp. 1–4, doi: 10.1109/ICCSDET.2018.8821063.

[53] B. Srikanth, “Detecting Selfish Nodes in MANETs,” National Institute of Technology Rourkela, 2014.

[54] H. Yadav and H. K. Pati, “A Survey on Selfish Node Detection in MANET,” in *International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 2018, pp. 217–221, doi: 10.1109/ICACCCN.2018.8748420.

[55] M. Bounouni and L. Bouallouche-Medjkoune, “Adaptive Credit-Based Stimulation Scheme for Dealing With Smart Selfish Nodes in Mobile Ad Hoc Network,” in *2018 International Symposium on Programming and Systems (ISPS)*, 2018, pp. 1–5, doi: 10.1109/ISPS.2018.8379006.

[56] K. Rama Abirami and M. G. Sumithra, “Evaluation of Neighbor Credit Value Based AODV Routing Algorithms for Selfish Node Behavior Detection,” *Cluster Comput.*, vol. 22, pp. 1–10, 2018, doi: 10.1007/s10586-018-1851-6.

[57] S. B. B and I. K, “Detection of Selfish and Malicious Node in Mobile Ad-Hoc Network with NS-2 Using Chord Algorithm,” *Int. J. Eng. Technol.*, vol. 9, no. 2, pp. 466–471, 2017, doi: 10.21817/ijet/2017/v9i2/170902326.

[58] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, “An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs,” *IEEE Trans. Mob. Comput.*, vol. 6, no. 5, pp. 536–550, 2007, doi: 10.1109/TMC.2007.1036.

[59] M. V. Baseri and H. Fatemidokht, “Survey of Different Techniques for Detecting Selfish Nodes in MANETs,” *J. Mahani Math. Res. Cent.*, vol. 11, no. 2, pp. 45–59, 2022, doi: 10.22103/jmmrc.2022.18626.1180.

[60] K. Balakrishnan, J. Deng, and P. K. Varshney, “TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks,” in *IEEE Wireless Communications and Networking Conference, WCNC*, 2005, vol. 4, pp. 2137–2142, doi: 10.1109/wcnc.2005.1424848.

[61] G. Borboruah and G. Nandi, “A Study on Large Scale Network Simulators,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 6, pp. 7318–

7322, 2014.

[62] N. Purohit, R. Sinha, and K. Maurya, “Simulation Study of Black Hole and Jellyfish Attack on MANET Using NS3,” in *2011 Nirma University International Conference on Engineering: Current Trends in Technology, NUiCONE 2011 - Conference Proceedings*, 2011, pp. 8–10, doi: 10.1109/NUiConE.2011.6153239.

[63] T. R. Murgod and S. Meenakshi Sundaram, “A Comparative Study of Different Network Simulation Tools and Experimentation Platforms for Underwater Communication,” *Bull. Electr. Eng. Informatics*, vol. 10, no. 2, pp. 879–885, 2021, doi: 10.11591/eei.v10i2.1466.

[64] R. Bagrodia *et al.*, “Parsec: A Parallel Simulation Environment for Complex Systems,” *IEEE*, vol. 31, no. 10, pp. 77–85, 1998, doi: 10.1109/2.722293.

[65] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: A library for Parallel Simulation of Large-Scale Wireless Networks,” *Proc. Work. Parallel Distrib. Simulation, PADS*, pp. 154–161, 1998, doi: 10.1109/pads.1998.685281.

[66] Y. Pan, “Design Routing Protocol Performance Comparison in NS2 : AODV Comparing to DSR as Example,” pp. 1–14, 2005, [Online]. Available: https://www.semanticscholar.org/paper/Design-Routing-Protocol-Performance-Comparison-in-2-Pan/87a6c2b37ad39c36879a1870bdaebe5774a50f6e.

[67] M. Probert, “AWK, GREP and SED – Three VERY Useful Command-Line Utilities,” 2016. [Online]. Available: https://www-users.york.ac.uk/~mijp1/teaching/2nd_year_Comp_Lab/guides/grep_awk_sed.pdf.

[68] K. Prabha and R. Anbumani, “Performance Evaluation of Packet Delivery Ratio for Mobile Ad-hoc Networks,” *Int. J. Comput. Appl. Technol. Res.*, vol. 6, no. 6, pp. 306–310, 2017, doi:

10.7753/ijcatr0607.1007.

[69] A. B. Malany and V. R. S. R. C. Dhulipala, “Throughput and Delay Comparison of MANET Routing Protocols,” *Int. J. Open Probl. Compt. Math.*, vol. 2, no. 3, pp. 461–468, 2009, [Online]. Available: http://www.emis.ams.org/journals/IJOPCM/Vol/09/IJOPCM(vol.2.3.10.S.9).pdf.

[70] E. M. Abdulkader Salem, “Comparative Study and Performance Analysis of Routing Protocols for MANET,” *IOSR J. Comput. Eng.*, vol. 16, no. 6, pp. 25–32, 2014, doi: 10.9790/0661-16612532.

[71] N. I. Sarkar and W. G. Lol, “A Study of MANET Routing Protocols: Joint Node Density, Packet Length and Mobility,” *Proc. - IEEE Symp. Comput. Commun.*, pp. 515–520, 2010, doi: 10.1109/ISCC.2010.5546763.

[72] S. Dodke, P. B. Mane, and M. S. Vanjale, “A Survey on Energy Efficient Routing Protocol for MANET,” in *2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT) 161*, 2016, no. file:///F:/02 Research/03 OSVM Thesis/03 References/military.pdffile:///F:/02 Research/03 OSVM Thesis/03 References/military.pdf, pp. 160–164.

[73] S. Kumari and K. T. Sikamani, “Communication Based Clustering to Detect Selfish Nodes in MANET,” Indian J. Sci. Technol., vol. 8, no. 12, pp. 1–6, 2015, doi: 10.17485/ijst/2015/v8i.

APPENDIX

1. Developing AODV based on Proposed Method

In this section, AODV protocol is modified in order to implement the suggested method inside the required files.

Here are the AODV source code locations. It contains path **.cc**, **.h**, and **.o** files. The **.cc** and **.h** files are source files, whereas the **.o** file is the object file. The header (**.h**) files are used to determine the capabilities of a particular protocol or algorithm by defining the data members and member functions of a specific class. The functions described in the header files are implemented in the C source files.

In this instance, we will alter two files in the ns-2.35/aodv/ subdirectory, aodv.h and aodv.cc.

Step 1: Make an entry in aodv.h

Source File : **~ns-2.35/aodv/ aodv.h**

Include the following header line at the beginning of the aodv.h file.

```
#include <mobilenode.h>
```

In the aodv class, declare the following variables (in protected scope)

```
MobileNode * Nodeaddr; //to create a node
```

Step 2: Make an entry in aodv.cc constructor

Source File **: aodv.cc**

variables are initialized in the aodv.cc

```
double Energy;
double Residual_energy;
double IE = 100;
double Threshold;
int GRR;
int unsentMassege;
int SRR;
int selfish;
```

double CR;

Declare the following variables inside the aodv constructor, these following variables were initialized befor

GRR = 0;

SRR = 0;

unsentMassege = 0;

CR = 0.0;

Energy = 0.0;

Step 3: Make an entry in aodv.cc

A- To get the number of GRR and SRR for each node in the network, Where GRR: Get Route Request, SRR: SendS Route Reply. I will add GRR++ and SRR ++ to each of AODV::recvRequest() function and AODV::sendReply() function.

Where:

AODV::recvRequest(Packet *p): This function is the one that gets called whenever a node receives a packet of the REQUEST type.

void sendReply(nsaddr_t ipdst, u_int32_t hop_count,nsaddr_t rpdst, u_int32_t rpseq,u_int32_t lifetime, double timestamp): This function sends Reply messages.

B- In AODV protocol, the run time information is required during a forward of a packet. So AODV has a forward() function that handles the code related to the forwarding of packets. So the function from the mobilenode.h will be called or used inside the forward() function of AODV. In order to access the necessary functions from mobilenode.h, the following lines need to be included in the

void AODV::forward(aodv_rt_entry *rt, Packet *p, double delay) {

```
/**** This piece of code fetches the Energy Value of the node. *****/
Nodeaddr = (MobileNode*) (Node::get_node_by_address(index));
Residual_energy = Nodeaddr ->energy_model()->energy();
```

The equations for detecting selfish nodes are also written in

Forward(aodv_rt_entry * rt, Packet * p, double delay) {

```
/**** This code calculates the value of CR *****/
 unsentMassege = (float)GRR-(float)SRR;
CR = (((float)GRR - (float)unsentMassege) / (float)GRR) * 100;
/***This code calculates the value of Energy and threshold of energy *****/
Energy = Residual_energy ;
Threshold = ((IE - Residual_energy) / Residual_energy) * CURRENT_TIME;
/***This code defines the selfish nodes  according to the following
conditions*****/
if ((CR) < 30) {
        if ((Energy) <  Threshold && (Energy) > 10) {
           selfish = index; //if the nodes is selfish, then part C is called
        }
         else if ((Energy) < 10) {
        rtable.rt_delete(ih->daddr()); //Here the nodes will be isolated

                and call part D
          }
        }
```

C- In this part, the selfish node is improved in order to be able to participate in network activities to the maximum extent possible. After the selfish nodes are discovered in the previous step, the rate of the selfish node will be reduced in AODV::sendRequest()function, where the function void sendRequest(nsaddr t dst) is what's utilized to send request messages.

```
    if (rt->rt_dst == selfish) {
     rt->rt_req_cnt = rt->rt_req_cnt / 2 ;
// rt_req_cnt is the no. of times we did network-wide broadcast
   }
```

D- In this part, the selfish nodes that are deleted from the routing tables will be replaced with normal nodes (where the entries of the routing tables will be checked are not selfish nodes).

In AODV::recvReply()function, the following code has been added:

```
if(rt == 0) {
if (! (CR) <  30 && (Energy) < 10)
rt = rtable.rt_add(rp->rp_dst);
    }
```

When a node gets a packet of type REPLY, it calls this function, which is AODV::recvReply(Packet *p).

Step 4: Recompile NS-2 to incorporate the aforementioned modifications. The simple procedure for recompiling is :

i. Open the terminal go to the root directory **sudo –i** and go to the directory ns-2.35/.

ii. Provide the commands **./configure --with-tcl-ver=8.5**, then **make clean**, after that **make**. Finally **make install**

iii. Execute Wireless tcl (In the next Section, the tcl file will be explained in detail)

2. Building Tcl File for testing LELCR and DRSCN Schemes

In this section, Tcl scripts are built to simulate the proposed method in the NS-2 environment under the requirement simulation parameters. All required configurations and settings will be included in Tcl file.

Step 1: Create a simulator instance:

```
set ns [new Simulator]
```

Step.2: Configure trace support by opening the "test.tr" file by calling the

```
procedure trace-all
#Trace File Generation
set tracefile [open test.tr w]
#Tracking every event and configuration
```

$ns trace-all $tracefile

Step 3: Build a topology object that monitors the location of mobile nodes inside the topological boundary and stores this information.

set topo [new Topography]

Step 4: Provide the topographical object's x and y coordinates.

$topo load_flatgrid $val(x) $val(y)

Step 5: Defining a General Operations Director (GOD)

create-god $val(nn)

Step 6: Configure nodes before building them. Node configuration API may include flat/hierarchical addressing, ad-hoc routing protocol, Link Layer, MAC layer, and IfQ.

Step 7: Provide initial node positions.

```
#set position of node
$ns  initial_node_pos $node()  50
```

Step 8: Configure the node mobility by mobi.tcl file as follows:

This API modifies node speed and direction.

```
set ns [new Simulator]
set mobi [open mobility.tcl w]
puts $mobi  "\$ns at $time \"\$node($i) setdest $x $y 10\""
```

Step 9: Set up UDP traffic between nodes.

set udp [new Agent/UDP]

Step 10: Define the stop time when the simulation concludes.

Procedure stop{} is called to flush out traces and close the trace file.

Step 11: Lastly, the command to initiate the simulation displays

"Starting Simulation...\n" $ns_ run

Therefore, these 11 steps may complete a one-time simulation

الخلاصة

الشبكات المخصصة للأجهزة المحمولة (MANETs) هي أنظمة مترابطة من العقد اللاسلكية التي تتواصل عبر روابط لاسلكية محدودة النطاق الترددي. تشارك العقد في MANETالمعلومات مع بعضها البعض من خلال العديد من العقد الوسيطة. ومع ذلك ، في بعض الحالات ، قد لا تشارك العقد في عملية التوجيه بشكل صحيح لأنها متباعدة جدًا أو ليس لديها طاقة كافية. هذا يجعل العقد تعمل بطريقة أنانية. هذه العقد (العقد الأنانية) تهتم فقط بنفسها، ولا تقوم بارسال البيانات الى العقد الاخرى المتصلة معها. قدمت جميع الدراسات الحالية طرقا للتحكم في العقد الأنانية من خلال الكسف عنها وعزلها عن بقية الشبكة. حتى الآن، لا توجد دراسة تحاول تحسين اداء العقد الأنانية وجعلها تشارك في أنشطة الشبكة. طورت هذه الدراسة طريقة جديدة للتعامل مع العقد الأنانية واستغلالها على أكمل وجه بدلاً من عزلها. في هذه الدراسة، الطريقة المقترحة تشمل ثلاثة مخططات. تم اقتراح المخطط الأول المسمى (Least Energy and Least Communication Ratio) من أجل اكتشاف العقد الأنانية بناءً على عاملين: الطاقة ونسبة الاتصال. المخطط الثاني هو مخطط العزل (Isolate Selfish Node) ، وهو المسؤول عن عزل العقد الأنانية وفقًا لمقدار الطاقة المتبقية. المخطط الثالث الذي يمثل المساهمة الرئيسية لهذا العمل المسمى (Detection, Reintroduced, and Collaborative of Selfish Node)، والذي يقترح تحسين سلوك العقدة الأنانية بعد اكتشافها عن طريق تقليل معدل الاتصال بالعقد الأنانية وفقًا لطاقتها الخاصة. تم استخدام محاكي NS-2 لتصميم محاكاة شاملة من أجل تقييم الطريقة المقترحة. توضح نتائج المحاكاة أن الطريقة المقترحة أدت إلى زيادة الإنتاجية بنسبة (٨٪) و (٤٤٪) وكذلك نسبة توصيل الرزم بنسبة (٦٦٪) و (٦٣٪) على التوالي. كما قللت من معدل إعادة إرسال الحزم بنسبة (٣٥٪) و (٣٥٪) ؛ تأخير بنسبة (٨٠٪) و (٧٩)؛ استهلاك الطاقة بنسبة (٨٦٪) و (٨٢٪) على التوالي مقارنة بالأعمال ذات الصلة (SNRRM و EBCS) في حالة اختلاف عدد العقد. وبالمثل ، في حالة تغيير السرعة ، أثبتت الطريقة المقترحة فعاليتها. كنتيجة نهائية ، حققت الطريقة المقترحة الغرض منها.

بسم الله الرحمن الرحيم

(اقْرَأْ وَرَبُّكَ الْأَكْرَمُ الَّذِي عَلَّمَ بِالْقَلَمِ عَلَّمَ الْإِنسَانَ مَا لَمْ يَعْلَمْ)

سورة العلق/ الاية ٣-٥

صدق الله العلي العظيم

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل-كلية تكنولوجيا المعلومات
قسم شبكات المعلومات

طريقة الكشف وإعادة التقديم التعاوني لمعالجة العقد الانانية في MANET

رسالة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل والتي هي جزء من متطلبات الحصول على درجة الماجستير في تكنولوجيا المعلومات - شبكات المعلومات

سناء جعفر حسن علي

بأشراف

أ.م.د رائد نصر كاظم خليل

١٤٤٤هـ ٢٠٢٢ م